

DESIGN AND INTERPRETABILITY OF CONTOUR LINES FOR VISUALIZING MULTIVARIATE DATA

A Thesis Submitted to the
College of Graduate and Postdoctoral Studies
In Partial Fulfillment of the Requirements
For the Degree of Masters
In the Department of Computer Science
University of Saskatchewan
Saskatoon

By
Gazi Md. Hasnat Zahan

©Gazi Md. Hasnat Zahan, 10/2020. All rights reserved.

Unless otherwise noted, copyright of the material in this thesis belongs to the
author

PERMISSION TO USE

In presenting this thesis/dissertation in partial fulfillment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis/dissertation in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis/dissertation work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis/dissertation or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis/dissertation.

Requests for permission to copy or to make other use of material in this project in whole or part should be addressed to:

Dean
College of Graduate and Postdoctoral Studies
University of Saskatchewan
116 Thorvaldson Building, 110 Science Place
Saskatoon, Saskatchewan S7N 5C9
Canada

Or
Head of the Department of Computer Science
176 Thorvaldson Building, 110 Science Place
Saskatoon, Saskatchewan, S7N 5C9
University of Saskatchewan
Canada

ABSTRACT

Multivariate geospatial data are commonly visualized using contour plots, where the plots for various attributes are often examined side by side, or using color blending. As the number of attributes grows, however, these approaches become less efficient. This limitation motivated the use of glyphs, where different attributes are mapped to different pre-attentive features of the glyphs. Since both contour plot overlays and glyphs clutter the underlying map, in this paper we examine whether contour lines, which are already present in map space, can be leveraged to visualize multivariate geospatial data.

We present five different designs for stylizing contour lines, and investigate their interpretability using three crowdsourced studies. We evaluated the designs through a set of common geospatial data analysis tasks on a four-dimensional dataset. Our first two studies examined how the contour line width and the number of contour intervals affect interpretability, using synthetic datasets where we controlled the underlying data distribution. Study 1 revealed that the increase of width improves the task performance in most of the designs, specially in completion time, except some scenarios where reducing width does not affect performance where the visibility of the background is critical. In Study 2, we found out that fewer contour intervals lead to less visual clutter, hence improved performance. We then compared the designs in a third study that used both synthetic and real-life meteorological data. The study revealed that the results found using synthetic data were generalizable to the real-life data, as hypothesized. Moreover, we formulated a design recommendation table tuned to give users task- and category-specific design suggestions under various environment constraints. At last, we discuss the comparison between the lab and online versions of study 1 with respect to display size (lab study was done on big screen and vice versa).

Our studies show the effectiveness of stylizing contour lines to represent multivariate data, reveal trade-offs among design parameters, and provide designers with important insights into the factors that influence multivariate interpretability. We also show some real-life scenarios where our visualization approach may improve decision making.

ACKNOWLEDGEMENTS

Firstly, I would like to thank my supervisors, Debajyoti Mondal and Carl Gutwin, for their valuable guidance. I am also grateful for the funding support from the university under the GWF (Global Water Futures) project that led me to this wonderful piece of work.

I am very much thankful to my wife and my daughter for giving me the courage to finish what I started. I am lucky to have some really good friends and lab mates who supported me all the way through.

Most importantly, I am eternally indebted to my parents, who taught me to accept failures, keep dreaming of success and never give up on something just because it is difficult.

Dedicated to my caring mother (Morzina Hoque), kind father (A K M Fazlul Hoque), lovely wife (Tazmin Khanam) and precious daughter (Ruwaiza Mehroz)

CONTENTS

Permission to Use	i
Abstract	ii
Acknowledgements	iii
Contents	v
List of Tables	viii
List of Figures	ix
List of Abbreviations	xi
1 Introduction	1
1.1 Motivation	1
1.2 Research Question	3
1.3 Contribution	4
1.4 Thesis Organization	5
2 Related Work	7
2.1 Preliminaries: Categorization and Terminology	7
2.1.1 Contour Plot	7
2.1.2 Color Blending	9
2.1.3 Visual Encoding	10
2.1.4 Pre-attentive Processing	10
2.1.5 Glyph	10
2.1.6 Normal Distribution	10
2.2 Cartographic Maps and Contour Plots	11
2.3 Multivariate Glyph Based Visualization	12
2.4 Stylization of Lines and Boundaries	13
3 Dataset Description	17
3.1 Synthetic Data	18
3.1.1 Rationale	18
3.1.2 Designing	19
3.2 Real-Life Meteorological Data	21
3.2.1 Differences between Data Types	21
4 Visual Encoding	22
4.1 Design 1: <i>Parallel Lines</i>	23
4.2 Design 2: <i>Color Blending</i>	24
4.3 Design 3: <i>Pie</i>	25
4.4 Design 4: <i>Thickness-Shade</i>	25
4.5 Design 5: <i>Side-by-Side</i>	26
5 Real World Examples	30
5.1 Detecting Variable Values	30
5.2 Trend Analysis	33
5.2.1 Trend: Temperature	33

5.2.2	Trend: Soil Moisture	34
5.3	Front Visualization	35
6	Implementation Details	38
6.1	Display Initialization	38
6.2	JSON Data Input and Processing	39
6.3	Contour Creation	39
6.3.1	Contour Simplification of Synthetic Data	39
6.4	Choosing A, B, C, and D	40
6.5	Visualization	41
6.5.1	Encoding <i>Parallel Lines</i>	41
6.5.2	Encoding <i>Color Blending</i>	43
6.5.3	Encoding <i>Pie</i>	43
6.5.4	Encoding <i>Thickness-Shade</i>	43
6.5.5	Encoding <i>Side-by-Side</i>	43
6.6	Evaluation	44
7	Study 1 (Contour Width)	45
7.1	S1: Participants, Data, and Tasks	46
7.1.1	Choosing the Tasks	46
7.2	S1: Hypotheses	47
7.3	S1: Procedure	47
7.4	S1: Results	48
7.4.1	S1: Interpretation tasks	49
7.4.2	S1: Background Task	49
7.4.3	S1: Subjective Effort and Preference	49
7.5	S1: Discussion	50
7.6	S1: Summary	50
8	Study 2 (Contour Intervals)	55
8.1	S2: Participants, Data, and Tasks	55
8.2	S2: Hypotheses	55
8.3	S2: Procedure	55
8.4	S2: Results	56
8.4.1	S2: Interpretation tasks	56
8.4.2	S2: Background Task	57
8.4.3	S2: Subjective Effort and Preferences	57
8.5	S2: Discussion	58
9	Study 3 (Design Comparison)	60
9.1	S3: Participants, Data, and Tasks	60
9.2	S3: Procedure	60
9.3	S3: Results	61
9.3.1	S3: Subjective Effort and Preferences	62
9.4	Overall Discussion	62
9.4.1	Task-wise Analysis	65
9.5	Design Recommendations	67
10	Study Insights	70
10.1	Display Size and Study Type	70
10.2	Study 1 Lab Version	70
10.3	Effect of Display Size on Visualization	72
10.4	Lab Study VS Online Study	72
10.5	Limitations and Future Work	72

11 Conclusion	75
References	77

LIST OF TABLES

4.1	Visual encoding table: <i>Parallel Lines</i>	23
4.2	Visual encoding table: <i>Color Blending</i>	24
4.3	Visual encoding table: <i>Pie</i>	26
4.4	Visual encoding table: <i>Thickness-Shade</i>	27
4.5	Visual encoding table: <i>Side-by-Side</i>	28
7.1	Different widths for Study 1	45
7.2	Tasks with domains for Study 1	46
7.3	Study 1 hypotheses	48
7.4	Study 1: NASA-TLX scores	50
7.5	Width choice after Study 1	50
8.1	Study 2 hypotheses	56
8.2	Study 2: NASA-TLX scores	57
9.1	Tasks with domains for Study 3	61
9.2	Study 3: design preference survey	63
9.3	Study 3: NASA-TLX scores	65
9.4	Task-wise analysis table: Completion Time	65
9.5	Task-wise analysis table: Accuracy	66
9.6	Design recommendation table	67
10.1	Lab VS online study hypotheses	70
10.2	Completion Time Chart	73

LIST OF FIGURES

1.1	Topography of Maungawhau - a classic volcano dataset. The contour bands' colors go from green (low) to yellow (middle) to white (high) and the contour lines are colored from white (low) to blue (high).	2
1.2	Research Scenario	3
1.3	Research Question and Goal	4
2.1	Visualizing scientific data using vector plots and contour plots in bump map [28]	8
2.2	Overlaying the UV vorticity (the curl of the horizontal velocity) (blue-red) over theta (temperature)(black-white). The temperature values can be seen through completely in the regions where the UV vorticity values are low [35]	8
2.3	Marching squares algorithm illustration by Nicoguaro, licensed under CC BY 2.0	9
2.4	(a) Two different glyphs for multivariate data [75] (b) visualizing 3 variables using line glyphs [47]	11
2.5	Visualizing 2 variables over Europe: average temperature in color and precipitation in height [95]	11
2.6	Visualizing 4 variables of a hurricane data (top 4) in different 2-variate and 4-variate views. Color blending wind velocity and water vapor (bottom left), 1 variable as background (blue scale for wind velocity) and 1 variable as ribbons (purple ribbons for water vapor) (bottom middle), and using ribbons for all 4 variables (bottom right) [101]	11
2.7	(a) Glyph-placement by the authors (top row) vs grid-based method (bottom row) [75] (b) visualizing open-ocean plankton density, current strength, and sea surface temperature with color, height, and density respectively [47] (c) a 4-variate map of US using color, orientation, size and density of a triangular glyph [46] (d) visualizing urban network centralities on a part of ShenZhen, China [55] (e) traditional Visualization (top) vs simulated brush strokes (bottom) varying color and texture [48] (f) the original 6,187 car trajectories (left) vs aggregated trajectories (right) [1] (g) visualizing age data over a region at multiple levels of spatial aggregation [106] (h) visualizing time and speed [83]	15
2.8	Visualizing crimes in West Lafayette and Lafayette, Indiana, where the blue line represents the Wabash river: (a) each point is a crime event (b) aggregating all the points by areal units (c) continuous domain approximation from point sampling (d) continuous domain approximation using solid lines on to the roads (e) authors' abstraction using a series of bristle lines applied along the roads [59]	16
2.9	Showing a bubble treemap of the S&P 500 index, decomposed into sectors and companies. Each circle represents a stock, its area \propto the mean closing price, whereas the standard deviation is depicted using the outlines [40]	16
2.10	Visualizing the distribution of Starbucks and McDonald's shops per 100,000 residents in the US with a per-capita bivariate cartogram [78]	16
3.1	Synthetic data - A, B, C, and D scatter plots. Each set of concentric circles denotes a cluster. Each colored band indicates a contour band.	17
3.2	Synthetic data - B, C, and D contours with background. The shades indicate the values of contour bands- darker (lighter) shade means higher (lower) value.	18
3.3	Synthetic data - final output	19
3.4	Real-life meteorological data overview	20
4.1	Visualizations using <i>Parallel Lines</i>	22
4.2	Visualizations using <i>Color Blending</i>	25
4.3	Visualizations using <i>Pie</i>	27
4.4	Visualizations using <i>Thickness-Shade</i>	28
4.5	Visualizations using <i>Side-by-Side</i>	29
5.1	Distribution of 4 variables over the Great Bear lake of Canada (Date: April 1, 2015 00:00AM). The contour bands are color filled according to values from no color (low) to darker shades of red (high).	30
5.2	Multivariate visualization with (left) glyphs occludes the map, and (right) grid stylization lacks the gradient information	31

5.3	5 different visualizations of the same data from figure 5.1	32
5.4	Trend: monthly average temperature- September, October, and November 2017	34
5.5	Trend: monthly average soil moisture- May, June, and July 2020	35
5.6	Front detection using <i>Parallel Lines</i> (right) and comparing with the actual NOAA data (left))	36
6.1	A closer look at the designs (the bottom right parts) with their respective legends. <i>Parallel Lines</i> , <i>Pie</i> and <i>Side-by-Side</i> have symmetric encoding scheme, unlike <i>Color Blending</i> and <i>Thickness-Shade</i>	38
6.2	Synthetic data - contour simplification	40
6.3	Illustration for the implementation details for different designs	42
6.4	Line break (for B in red) in <i>Parallel Lines</i> (at line segment intersections) due to drawing perpendicular lines along a contour line	44
7.1	Illustration for interpretation (task 1-4) and background (task 5) tasks	51
7.2	Study Procedure	52
7.3	Study 1: width VS accuracy and completion time	53
7.4	Study 1: background task	54
7.5	Study 1: width preference survey	54
7.6	Study 1: NASA-TLX scores	54
8.1	Study 2: contour intervals VS accuracy and completion time	56
8.2	Study 2: contour intervals preference survey	57
8.3	Study 2: NASA-TLX scores	57
8.4	Study 2: task accuracy for the five designs	59
9.1	Study 3: Two additional tasks	62
9.2	Study 3: overall performance of the different designs	63
9.3	Task performances in Study 3, for (top) different designs, and (bottom) different datasets	64
9.4	Study 3: design preference survey	68
9.5	Study 3: NASA-TLX scores	69
10.1	Design-wise comparison between lab and online study	71

LIST OF ABBREVIATIONS

AMT	Amazon Mechanical Turk
SVG	Scalable Vector Graphics
RM	Repeated Measure
ANOVA	Analysis of Variance
GPU	Graphics Processing Unit
NASA-TLX	NASA Task Load Index

CHAPTER 1

INTRODUCTION

The increasing availability of data sources and sensors has lead to various multivariate data visualization techniques. Multivariate data visualization [23, 72] aims at conveying multiple data sets to users by visualizing multiple attributes in a common visual space. Researchers have been trying ways to visualize data on small [17], big and multiple [35] screens, considering the challenges of space constraint while visualization.

The human brain perceives different visual cues like color, hue, luminance, texture, size, shape, orientation, motion in different ways. While the comprehension of data values from the cues greatly affects the completion time and accuracy of tasks, the learning curve also varies from encoding to encoding, some being quite intuitive in nature. Thus, the research on the human perception of visual cues [45, 49] has played a very important role in designing visualization systems.

In the past years, several visualization ideas have been closely evaluated [14] through user studies to develop efficient designs. Visualization has been deployed in various sectors of interest e.g. meteorology [74], medical analysis [85], flow visualization [61], business [91], sports [63] etc. A large body of systems greatly depends on interactivity [19, 19, 21, 24, 35, 46, 52, 63, 90, 91, 94] where users get to interact with the system to get their desired visualization, while some systems incorporate human input [54] in the implementation process to get the expected output. In this thesis, we explore multivariate data visualization for geospatial datasets.

1.1 Motivation

Contour plots are widely used to visualize geospatial information on two-dimensional maps. Common examples include elevation visualization on topographic maps [69, 71, 90], travelling distance plots on isochrone maps [79], and graphical plots in thermodynamics [68]. Contour lines and contour intervals are two important features of a contour plot. A contour line (isoline) represents a fixed threshold value and connects the map points having that value. A contour interval corresponds to a range of values within the bounds indicated by two successive threshold values. Figure 1.1 illustrates a contour plot with 10 contour intervals. Each contour interval is mapped to a color scale from green to yellow to white and contour lines are mapped from white to blue. Indicated by the solid directed line in red, the value gradually increased from the border to the middle. The line starts from a green contour band bounded by white contour lines, representing the lowest value, and ends in the white contour band with the darkest blue contour line, representing the highest value. Similarly, the dashed red line indicates a continuous decrease in value from the middle of the map to the

border.

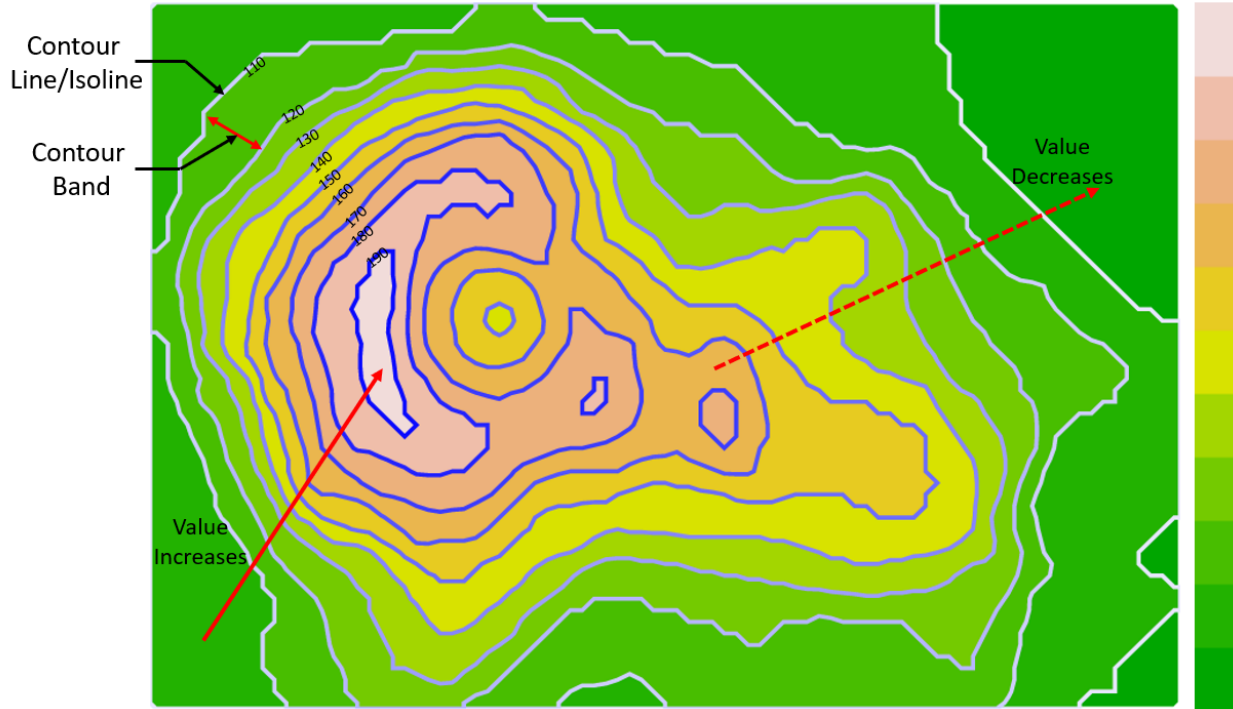


Figure 1.1: Topography of [Maungawhau](#) - a classic volcano dataset. The contour bands' colors go from green (low) to yellow (middle) to white (high) and the contour lines are colored from white (low) to blue (high).

The simplicity and rich information found in contour plots make them a popular choice in many geospatial data analysis [10, 31, 76] tasks. However, the effectiveness of individual contour plots becomes limited for multivariate data analysis [25, 33, 36]. For example, consider finding regions with high temperature and low precipitation (drought-prone areas [29]) from two separate contour plots for temperature and precipitation. The common solution for such tasks is to use bivariate choropleth maps [64], or blending the color of the two contour plots [42, 43, 53]. While the former creates sharp region boundaries, the latter helps to understand the transitional zones [89].

Scenarios that require analyzing combinations of three or more attributes are ubiquitous in geospatial data analysis. For example, understanding coastline erosion requires relating historical erosion with coastal properties such as slope, wave energy, tide range, sea-level change, as well as their morphological classifications [26]. Investigating climate conditions on crop production, studying and predicting hurricane paths [27, 74], estimating the vulnerability of glacial boundaries [41], understanding how the change in one climatic variable impacts the others [67], are examples where interaction among multiple attributes is important for decision making.

Visualization with small multiples [2] and color blending contour plots in pairs can assist the interpretation of multivariate data, but this increases both mental workload and interaction difficulty. Multivariate visualizations that encode data attributes into different pre-attentive perceptual features of a visual element (glyph) [6, 86, 99] such as size, shape, color, and texture, are typical ways to visualize geospatial information on a map. A well-known limitation of a glyph-based visualization is that it clutters the map [19]. While a dense overlay occludes the view of the base map, a

sparse overlay compromises the perception of geospatial connectedness.

1.2 Research Question

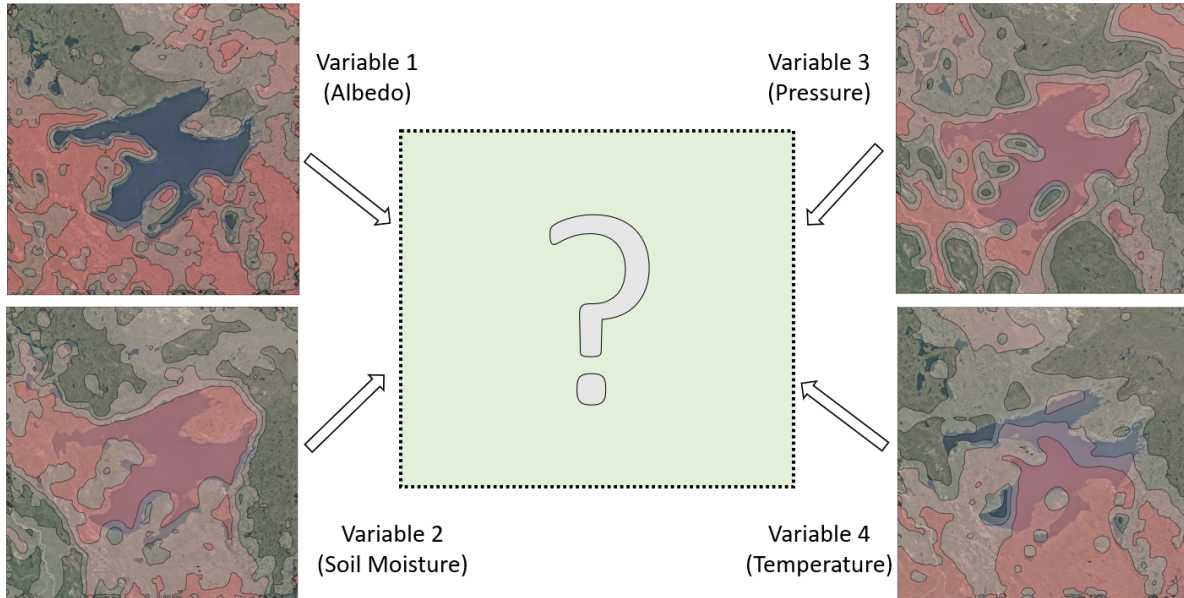


Figure 1.2: Research Scenario

In a real-life scenario (Figure 1.2) where we have 4 geospatial variables to visualize on an interface, the common approaches that use contour overlays or glyphs, suffer from visual clutter. We wanted to find a potential visualization alternative with good interpretability and task performance. We wanted to explore the following research question:

‘Can we leverage the points on the contour lines of one variable to show the other data attributes effectively?’

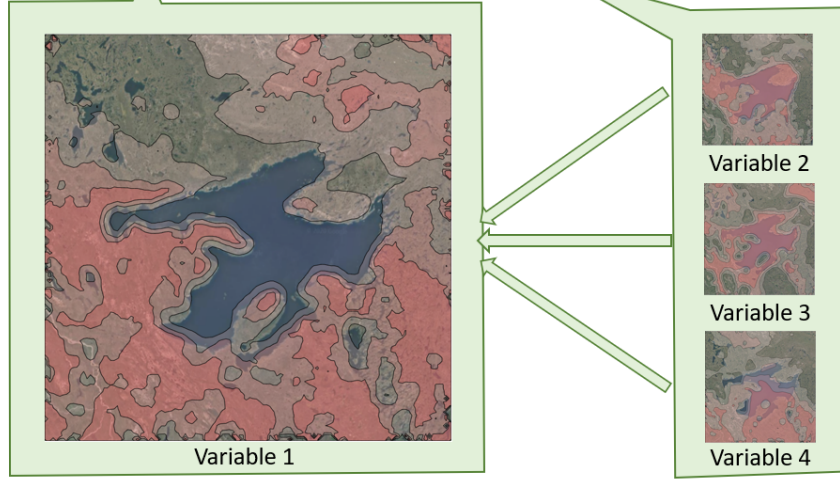
Figure 1.3 shows both the research question and our goal, which is to examine whether we can-

Leverage the contour lines of one variable to display the other three, minimizing visual clutter and base map overlap.

We consider geospatial data with four attributes: A, B, C, and D, and encode B, C, and D along the contour lines of A. We design five visual encodings and investigate whether users can interpret the attribute values (high, low), trends (increasing or decreasing), and relationships (similar or opposite trends) along a contour line, or across a set of contour lines. These interpretations encompass the most common geospatial data visualizations tasks involving one or multiple variables. For example, finding regions with different values of A will lead to peak or valley identification. Checking B values across multiple contour lines of A will show where the trend direction is. Comparing C and D values across the contour lines of A will reveal if they are positively or negatively correlated to each other and can also show their rate of change. Moreover, observing value changes along a contour line will reveal the dependency of that attribute on the others where A remains the same. Such tasks are very common in data visualization and sometimes very crucial in various decision-making scenarios.

Research Question

Can we leverage these points along the contour line to show the other data attributes effectively?



Research GOAL

Leverage the contour lines of one variable to display the other three
[minimize visual clutter and base map overlap]

Figure 1.3: Research Question and Goal

A rich body of research focuses on stylizing linear elements on a map (e.g., road networks, trajectories, and region boundaries) to encode multiple attributes of the region of interest [46, 59, 82, 106]. Since our designs do not depend on the relationship between the contours of B, C, and D, they can also be used for such lines and boundaries, as well as for regular grid overlays, to reduce visual clutter. However, the inherent connection of a contour plot with a three-dimensional surface inspires many trend analysis tasks that are not common in region or trajectory analysis contexts.

1.3 Contribution

In this thesis, we propose five designs to analyze multivariate geospatial data with four attributes (A, B, C, D) and leverage the contour lines of A to encode the others. Since the encoding position of B, C, D is determined by A's contour line, users may want to vary the number of contouring thresholds for A or use a different base contour plot.

Since the real-life dataset may not capture all possible attribute combinations, we also describe how to design a synthetic dataset to examine the influence of various design parameters through controlled experiments.

We conduct three crowdsourced studies that evaluate our designs. The first two studies reveal how the contour line width and the number of contour intervals influence the visual interpretability of our designs. The third study uses both synthetic and real-world meteorological datasets to explore how the completion time and accuracy of common geospatial data analysis tasks vary based on designs and choice of datasets.

In addition to revealing insights into our designs, our experimental results also suggest that results obtained using

synthetic datasets generalize to real-world datasets. The insights gained through these studies will inform and inspire future research on curve stylization for geographical visual analytics.

1.4 Thesis Organization

The thesis is divided into several chapters in order to describe the whole execution step by step, starting with the description of some related works, then the design implementation, user studies, and results, and ending with an overall summary of the work.

Chapter two provides an overview of the related research works in visualizing multivariate data. Since the thesis is aimed to augment the contour lines with glyphs, the chapter divides the related works into 3 sections- cartographic maps and contour plots, multivariate data visualization using glyphs, and stylization of lines and boundaries. We discuss visualization techniques, data types, and the variables they visualize.

Chapter three describes the two types of datasets used for visualization- synthetic and real-life meteorological data. For synthetic data, the process of generating clusters for 4 variables, assigning values to the clusters to cover all possible inter-variable combinations, and finally generating 2 sets of data is discussed. For real-life data, the process of taking a $200 \times 200 \times 1$ slice from a 699 (latitude) \times 639 (longitude) \times 24 (timestamp) weather data with 4 variables (Temperature, Pressure, Soil Moisture, and Albedo) and then, filtering outliers, normalizing and smoothing the data set using a 5×5 mean filter to finally generate 2 sets of data is discussed.

Chapter four describes our five design encoding schemes in detail. Each encoding description includes the encoding or glyph type, position, color, the minimum and maximum possible value of the encoding with demonstrations using both synthetic and real-life data.

Chapter five demonstrates a real-world scenario of how our designs can help the users to easily find out some variable values, trends and comparisons, etc. The section starts with the basic contour map of all 4 variables and then visualizes all variables in our five designs. Each design shows its strength to complete some common geospatial data analysis tasks with ease.

Chapter six describes the deep implementation procedure explaining the steps from display initialization till visualization. Display initialization and JavaScript Object Notation (JSON) data processing talk about setting up the Scalable Vector Graphics (SVG) width and height according to each design(to try to maintain the equal number of pixels), creating contour plots using *d3.contours()*, and balancing the number of pixels in all five designs. Contour creation describes the output of the *d3.contours()* function and simplification of contour lines for the synthetic data. The last step is visualization, which demonstrates visualizing data on the contour lines in higher detail using illustrations.

Chapter seven describes the first of the 3 crowdsourced studies, which aims to investigate the effect of contour width on design interpretability. The trade-off here is - a higher width should lead to easier interpretation and higher visual clutter and higher background overlap. Therefore, we wanted to check how the users perform (Study 1) in both of the interpretation tasks (where the user does not need to consider the underlying background map) and background related tasks (does not need to consider the visualization layer above the map). The study had a total of 57 tasks (4 different

task domains except for background task) with 12 different width choices, and was run on 127 participants through the Amazon Mechanical Turk (AMT) platform. Repeated Measure (RM) Analysis of Variance (ANOVA) on the results shows that width had a significant effect on performance. Out of 2 hypotheses for the study, both got partially supported by the result data.

Chapter eight describes Study 2, aiming to investigate the number of contour intervals on design interpretability. The trade-off is similar to Study 1 - a higher number of contour intervals should lead to easier interpretation but increase visual clutter as well. So we checked 2 more hypotheses using 69 tasks (4 different task domains except for background task) on 68 participants. Similar to Study 2, there were both interpretation and background tasks to be completed through AMT. RM ANOVA of the results shows partial support of one of the two hypotheses for the study.

Chapter nine describes the last study (Study 3) which is aimed to create a task and environment-specific design recommendation table to help users decide the appropriate data visualization techniques to use for a specific task domain, under a specific environment. Both synthetic and real-life data is used to run the study. We collected 78 full responses, each having 60 tasks (6 different task domains). Finally, a design recommendation table is constructed.

Chapter ten deals with the effect of display size and study type (lab or online study) on performance. Study 1 was run both in a lab setting and on Amazon Mechanical Turk (AMT) on a different set of participants. The goal was to check whether the result supports the last 2 hypotheses based on display size and study type. The results support one partially. More insight can be revealed in the future if the same study can be run on an equal number of participants in both settings. Finally, some limitations to the current approach are pointed out, along with future works to deal with them. The section talks about mapping to larger maps, Graphics Processing Unit (GPU) implementations, adaptive choice of contour intervals, additional ways to visualize data at regions with fewer contour lines, systematic implementation giving users the flexibility to choose the color, contour width and more interactions. Moreover, running controlled studies in a lab setting, focus groups with meteorological experts can be considered for gaining more insights into this research.

Chapter eleven concludes the thesis with a summary of the whole research including our findings and contribution.

CHAPTER 2

RELATED WORK

Visualization of multidimensional data [38] has spawned a rich body graphical charts (e.g., radar chart, iconography), interactive techniques (e.g., scatterplot matrix, parallel coordinates), and data embedding techniques (e.g., force layout, t-SNE). However, most geospatial data are typically visualized on a two-dimensional map to reveal spatial properties. In this chapter, we briefly review the literature on multivariate geovisualization techniques.

2.1 Preliminaries: Categorization and Terminology

This section will cover the basics of the most common terminologies related to multivariate geovisualization techniques, followed by the recent works. According to Keim and Kriegel [57, 58], multivariate data exploration techniques can be divided into 6 categories- geometric, icon-based, pixel-oriented, hierarchical, graph-based, and hybrid techniques. Geometric techniques focus on the geometric transformation and projection of data. Some geometric techniques include scatterplot-matrices, landscapes, projection pursuit techniques, prosection Views, hyperslice, parallel Coordinates, andrews Curve, and star coordinates [57]. Icon-based techniques visualize data as icons where the features of an icon encode the data value. They include chernoff-faces, stick figures, star glyph, shape coding, color icons, texture, and tileBars [57]. Pixel-oriented techniques encode data values in pixels. Some examples are space-filling curve, recursive pattern, spiral technique, axes techniques, circle segments technique, a combination of spiral, axes and color techniques [57]. Hierarchical techniques, as named, visualize data in hierarchy. Some include dimensional stacking, worlds-within-worlds, treemap, cone trees, and infoCube [57]. Graph-based techniques convey the underlying data distribution through line graphs. Two of such techniques are seeNet and narcissus [57]. And lastly, hybrid techniques are the combination of the former ones.

Among the techniques mentioned above, we discuss the related works on 3 areas- cartographic maps and contour plots, multivariate glyph-based visualization, and stylization of lines and boundaries. Before describing the details, we discuss some terminologies related to this thesis work.

2.1.1 Contour Plot

A contour plot is considered as a graphical representation of a 3-dimensional surface on a 2-dimensional plane, which is typically made from continuous spatial data, e.g., temperature data all over Canada. For a value for z (e.g. temperature), isolines connect the (x,y) coordinates attaining that z value. The isolines can be thought of as slices of x - y planes along

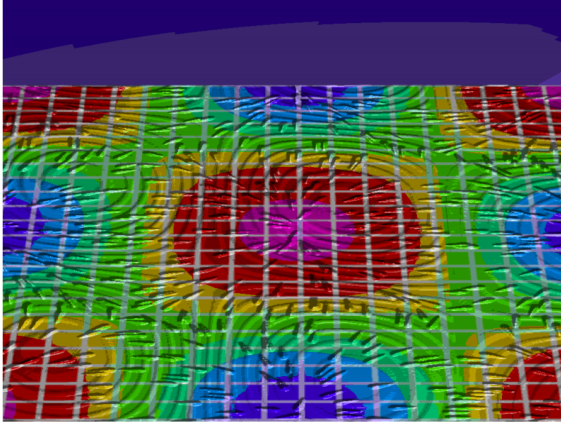


Figure 2.1: Visualizing scientific data using vector plots and contour plots in bump map [28]

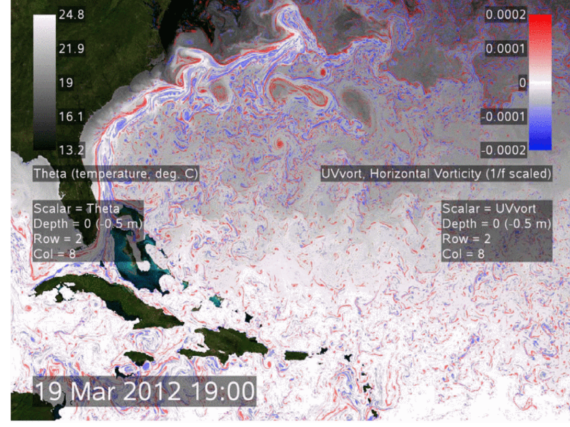


Figure 2.2: Overlaying the UV vorticity (the curl of the horizontal velocity) (blue-red) over theta (temperature)(black-white). The temperature values can be seen through completely in the regions where the UV vorticity values are low [35]

the z-axis. The contour map looks like a number of x-y slices stacked on top of each other. These types of plots are used in making topological maps. A common example would be an elevation map over a region of interest.

The javascript function used in this thesis to generate contours is [d3.contour\(\)](#). The function uses the Marching Square [73] algorithm to compute contour polygons. The algorithm works as follows: given a 2D scalar field (rectangular array of inputs) and a threshold value as two inputs, the function constructs a [GeoJSON](#) MultiPolygon geometry object which represents the area where the input values are greater than or equal to the given threshold value.

The marching square algorithm (Figure 2.3) divides the whole rectangular input into 2×2 pixel blocks to form contouring cells, and the cells aggregate to form the whole grid. Each cell is processed independently in 3 following steps.

Given an isovalue i (for Figure 2.3, $i = 1$), for each cell:

1. Build a binary index consisting of 4 bits for 4 corner points of the cell: walk around the cell in a clockwise direction appending 0 or 1 for the pixel values from the top left to the bottom left corner. Append 0 is the data value is above the isovalue, or 1 if the data value is below the isovalue. This way, a 4-bit index will form, having 16 possible values, ranging from 0 to 15.
2. Use the cell index as the case number and match it to a case from the pre-built lookup table with 16 cases, listing the edges needed to represent the cell (see the lower right part of Figure 2.3).
3. Use linear interpolation between the original data values and find the correct position of the contour line along the cell edges.

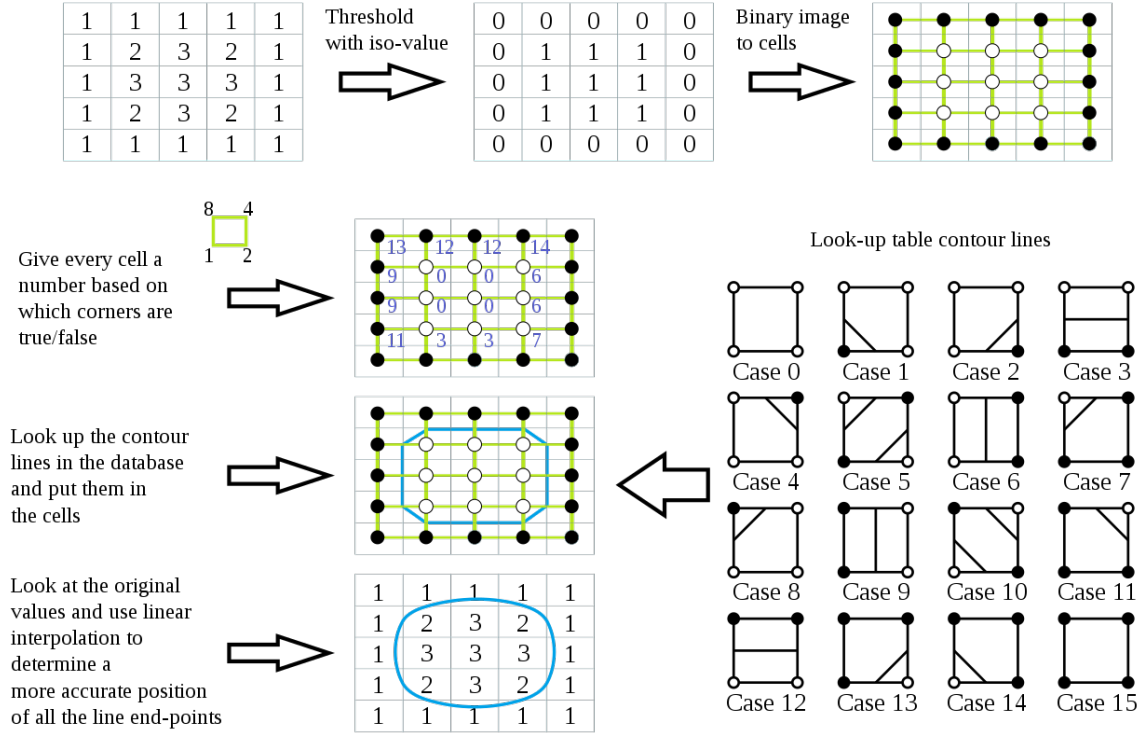


Figure 2.3: Marching squares algorithm illustration by Nicoguaro, licensed under CC BY 2.0

2.1.2 Color Blending

Blending colors is a common way to visualize information, e.g., temperature and pressure over a map. There are multiple **blending modes** to do so- each gives a different look to the underlying set of colors. Given 2 layers a and b , where a is the base layer and b is the top layer, the basic arithmetic blend modes- addition, subtract or divide, as they are named, adds, subtracts or divides the components of a and b , respectively. The difference mode subtracts the base layer a from the top layer b or the other way round, to always get a non-negative value. The darken mode takes the smallest 3 components of the a and b pixels. If a has the components (r_1, g_1, b_1) and b has the components (r_2, g_2, b_2) , the mode formula is $[\min(r_1, r_2), \min(g_1, g_2), \min(b_1, b_2)]$. On the other hand, lighten mode takes the biggest components i.e. the formula becomes $[\max(r_1, r_2), \max(g_1, g_2), \max(b_1, b_2)]$.

The dodge (screen, color dodge) and burn (multiply, color burn) modes change the lightness of the top layer, similar to the dodging and burning images in a dark room. Multiply simply multiplies ($f(a, b) = ab$) each component of a with b . Multiply mode usually darkens the top layer. Screen mode, on the other hand, brightens an image by inverting the two layers, multiplying, and then inverting again. The formula is $f(a, b) = 1 - (1 - a)(1 - b)$. Some comparative examples of different blending modes can be found at [77, 103].

2.1.3 Visual Encoding

Memory encoding is the process of converting an item of interest into a construct using cues, e.g. visual, auditory, sensory stimuli, etc. to store it in the brain as memory (short-term or long-term) for further retrieval [39]. There are two approaches to encoding memory- the physiological approach looks at how a stimulus is represented by neurons firing in the brain, and the mental approach looks at how the stimulus gets represented in the mind [81]. The mental approach can be of mainly four types- visual, elaborative, organizational, acoustic, and semantic. Visual encoding is the process of encoding images and visual sensory information into memory for understanding. The use of efficient visual encoding is one of the main foundations of great visualization. Humans perceive different visual features or cues (color, texture, luminance, size, shape, orientation, density, etc.) of a visual representation as different attributes of the underlying data. Multivariate data visualization techniques deal with multiple visual cue mappings to demonstrate multiple variables. Gestalt principles [93] are considered very important in understanding how human visual perception works, in order to design efficient visualization techniques.

2.1.4 Pre-attentive Processing

When we gather information from our surrounding environment through our senses, the first step involves the collection of all available information subconsciously, without the active presence of the conscious mind [96]. This step is called pre-attentive processing. Then the brain filters out the information, and attentive processing comes into play, which is responsible for selecting specific information for further complex processing. Some pre-attentive visual features include color intensity, hue, size, shape, density, orientation, and movement. The study of pre-attentive processing is important in various sectors, e.g. education, advertising, and research.

2.1.5 Glyph

In topography, a *glyph* is a symbol with a set of pre-attentive features mapped to the data variables to be visualized. Over a map of interest, the symbols are placed with varied feature set according to the underlying data value. Such features can be size, shape, color, hue, orientation, density etc. Glyphs are very commonly used in multivariate data visualization- 2D and 3D in various sectors including medical data visualization [85]. In Figure 2.4(a), we can see 2 types of glyphs placed on the maps, each in 2 different ways along a column. They have their size, shape varied according to the underlying data. Again, the glyphs of Figure 2.4(b) change their color, height, and density on the basis of 3 data variables.

2.1.6 Normal Distribution

In probability theory, a normal (or Gaussian) distribution is a continuous probability distribution for a real-valued random variable. The formula is

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

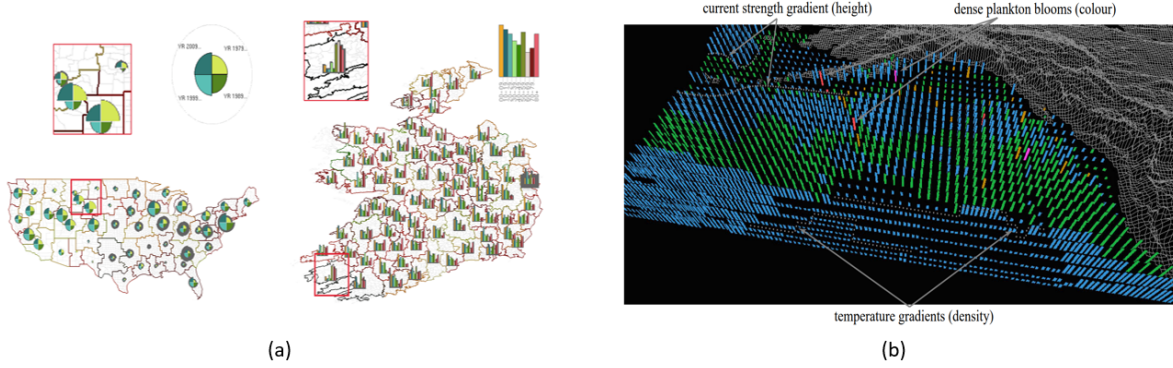


Figure 2.4: (a) Two different glyphs for multivariate data [75] (b) visualizing 3 variables using line glyphs [47]

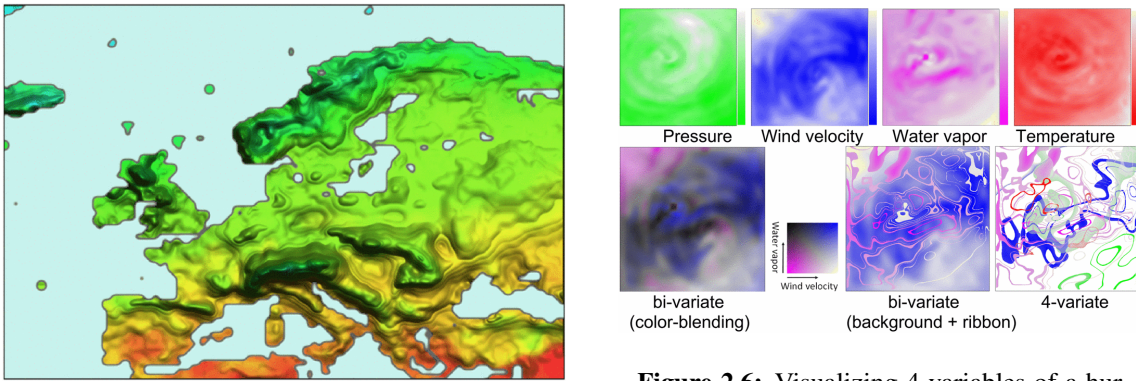


Figure 2.5: Visualizing 2 variables over Europe: average temperature in color and precipitation in height [95]

Figure 2.6: Visualizing 4 variables of a hurricane data (top 4) in different 2-variate and 4-variate views. Color blending wind velocity and water vapor (bottom left), 1 variable as background (blue scale for wind velocity) and 1 variable as ribbons (purple ribbons for water vapor) (bottom middle), and using ribbons for all 4 variables (bottom right) [101]

where μ is the mean, σ is the standard deviation and σ^2 is the variance of the random variable.

2.2 Cartographic Maps and Contour Plots

Geospatial data are often shown using choropleth maps [34, 64, 70], and contour plots [68, 69, 79, 90]. In a choropleth map, a set of predefined regions is colored or patterned to represent the value of a single variable. While choropleth maps and cartograms [32] reveal properties of a region, a contour plot helps to understand the data distribution on a map and find regions with similar properties. Traditionally, both these visualizations have been used to visualize univariate data. However, by using color blending or texture pattern, one can extend them to visualize bivariate data. Creating a high-quality bivariate choropleth or contour map requires careful choice of blending colors [7, 65] and textures [59]. Bivariate maps are popular, as they allow users to directly infer relationships between two variables. Data analysts on such maps often use color blending for finding probable correlations between two geospatial variables (Figure

2.2) [35]. Wijk and Telea [95] added ridges to the contour lines of the map (Figure 2.5) to create more perceptive bivariate visualization. Their goal was to demonstrate the increased performance of adding simple ridges i.e. shaded light fields to a scalar function of 2 variables can be leveraged in multiple real-life applications while having the ability to be extended to more variable mappings using color, texture or lines, etc. Crawfis and Allison [28] extended the use of textures into vector fields and contour plots (Figure 2.1) with raster operations to visualize scientific data.

Researchers have also attempted to construct trivariate choropleth maps using CMY color model [87]. Cao et al. [11] showed that 3-variate seismic data combinations can be analyzed visually by mapping data values to different color channels and then using traditional color models (e.g., RGB, CMY and HSV models). Wu and Zhang [101] examined a 4-variate map (Figure 2.6) that for each variable, captures the contour band information into thin visual ribbons, and then overlays the ribbons for all four variables using four different colors. For 3-variate maps, one can blend two variables and then overlay the third variable using a visual ribbon with a different color. Although such approaches are promising, they all clutter the base map and often appear to be difficult to interpret for three or more variables.

2.3 Multivariate Glyph Based Visualization

Symbol overlays on a map are a popular way to visualizing geospatial information [99]. For example, consider a scatterplot of Chernoff faces [9, 15], i.e., face icons representing the data points via cartoonish expressions, on a map. Glyphs can take different forms and shapes such as stick figures, vectors, icons, and even complex textures. However, glyphs are often designed to encode data into features that can be perceived through pre-attentive visual channels [100]. A rich body of visualization design research examines how humans perceive various combinations of geometric, optical, relational, and semantic channels. We refer the readers to the surveys [19, 38, 84, 99] for a detailed review of glyph design. Fuchs and Schumann [37] took a step further by overlaying visugrams (icons of complex diagrams such as ThemeRiver and TimeWheel) on a map.

Healey et al. [46] proposed a semi-automated system (Figure 2.7(c)) called visualization assistant (ViA), which takes data input and then collaborates with the users to provide the best possible set of visual mappings among the possible ones (color, size, orientation, hue, luminance, and density). The system combines a human's visual perception and an artificial intelligence algorithm to produce appealing visualizations. The 3 different visualizations of Figure 2.7(c) visualizes the same data in 3 different variable to encoding mapping alternatives.

Glyph based visualizations often come with a careful glyph positioning technique [52, 75, 101], as creating glyphs for many data points on a map creates overlapping. Kindlmann and Westin [60] proposed a glyph packing technique leveraging a particle system that creates a dense visual pattern and provides visual continuity similar to texture-based visualizations. They showed that such glyph packing in two dimensions can reveal large regions with similar structure, e.g., fiber structures. A number of interactive exploration technique has appeared in the literature to mitigate the overlapping problem and ease the interpretation of the glyph plots. Chung et al. [21] proposed high-dimensional, focus and context glyphs that are visually sortable. They observed that various sorting strategies on sortable glyphs can reveal trends in data. Interactions that allow users to pan, zoom and rotate a glyph plot have been proved to be useful

to mitigate the occlusion problem [75, 94]. Figure 2.7(a) shows the work by McNabb and Laramée [75], where they developed an algorithm pipeline that can adjust the level of detail of the visualization aided by a guided glyph placement technique. The authors’ method output (top row) provides a cleaner and better view than the corresponding grid-based method output (bottom row). The figure shows 3 datasets in columns- (left columns) Comparison of CCGs, (middle column) Comparison of US Counties, and (right column) Comparison of Ireland’s Electoral Divisions. Glyphs created in grid-based methods are often de-coupled from the geo-space they actually represent. The blue arrow shows the cause of the incomparable data values.

Apart from the study of glyph positioning, the visual perception of orderability of different visual mappings (size, hue, orientation, value, shape etc.) is also crucial for the user to get the idea of the data they get mapped to. Chung et al. [20] ran studies to examine the perceptual orderability of visual mappings and how they impact the performance of min and max judgements. They found out that the visual mappings that appear more ordered to the users usually improve their performance of min and max judgements.

Various texture metrics such as contrast, coarseness, periodicity, and directionality [66, 92] have been used to visualize multivariate data. Healey and Enns [47] introduced pexels that encode multi-dimensional datasets into multi-colored perceptual textures (Figure 2.7(b)) with height, density, and regularity properties. Low to high plankton densities are represented with blue, green, brown, red, and purple, taller pixels represent stronger currents, and denser pixels represent warmer temperatures.

An artistic representation of data can be more perceptible to human eyes by capturing attention through evoking emotional response. Healey et al. [48, 50] used simulated brush strokes to visualize geospatial data (Figure 2.7(e)) that conveys the underlying multivariate data to the user.

Shenas and Interrante [88] showed that a combination of color and texture can be used to meaningfully convey multivariate information with four or more variables on a choropleth map.

Some multivariate visualizations put special emphasis on lines and boundaries instead of cluttering the map with large overlays. The following section briefly reviews such design contexts.

2.4 Stylization of Lines and Boundaries

Stylized lines naturally appear in the visualization of trajectory data. For example, traffic flow data are often color-coded on the road networks as heatmaps [55, 98]. Andrienko et al. [1] extracted characteristic points from the car trajectories and aggregated them (Figure 2.7(f)) to create flows between cellular areas to reveal movement patterns in a city. They used stylization to depict various information about the aggregated flows. The original figure (Figure 2.7(f)(left)) has 6,187 car trajectories. The aggregated trajectories (Figure 2.7(f)(right)) visualized all flows visible on the original one, while the maximum arrow width is mapped to 253 trajectories. Huang et al. [55] modeled taxi trajectories using a graph (Figure 2.7(d)). They stylized the streets based on node centrality and overlaid rose charts to visualize other traffic information. The figure shows the most crowded 4 regions of the map using pagerank algorithm on a VW2 graph, where each vertex weight is the number of taxis passing this street in both directions. Charles et al. [3, 83] investigated

combinations of thickness, monochromatic color scheme, and tick mark frequency on a line to encode time and speed on a two-dimensional line (Figure 2.7(h)). They observed encoding speed with a color scheme, and time using one of the other two features to improve user perception. The figure has several encoding choices- (a) No encoding (b) size (or stroke width) represents speed; (c) color represents time elapsed; (d) color represents speed and size represents time elapsed; (e) segment length (spacing between ticks) represents time distribution, from which speed can be inferred (the closer two ticks, the slower); and (f) color represents speed on top of segment length. All encoding were visualized on both straight and curved lines.

Geographic cluster visualization and map generation techniques have also considered line stylization. Christophe et al. [18] proposed a pipeline for generating artistic and cartographic maps that integrate linear stylization, patch-based region filling and vector texture generation. Kim et al. [59] created Bristle Maps that put bristles perpendicular to the linear elements (streets, subway lines) of the map (Figure 2.8). To make the bristles continuous along the path, they approximated the information along the path using kernel density estimation, and then encoded multivariate information into the length, density, color, orientation, and transparency of the bristles. They noticed bristle map to outperform point maps (using color-coding) and line maps (using color and thickness) for multivariate encoding, but they also mentioned bristle map to be challenging to create because a poor choice on variable encoding can result in a suboptimal visualization.

Zhang et al. [106] introduced TopoGroups that aggregates spatial data into hierarchical clusters, and show the information about the geographic clusters on the cluster boundaries. They also used multiple visual encoding strategies to convey the hierarchical and statistical information of the clusters and used boundary distortion algorithms to minimize the visual clutter. Figure 2.7(g) mimics the work to provide an example scenario. The age summary of the population of a region has been visualized. Despite different distributions of age at different clusters, the highest level of aggregation shows more adults than children and more children than old people over the region.

Although TopoGroups summarizes the cluster information along the boundary, Zhang et al. noticed that the users may have the wrong interpretation that the visualization represents the local statistics near the boundary. In subsequent work, Zhang et al. [107] examined TopoText that replaces the boundaries and clusters using oriented texts.

Visual encoding of lines and boundaries have been widely used in visualizing data uncertainty [13, 40]. Cedilnik and Rheingans [13] overlaid a regular grid on the map and then stylized the grid edges using blur, jitter, and wave. Data uncertainty has also been mapped to contour lines, where uncertainty is mapped to line color, thickness, and dash frequency [80]. Line stylization has also been used in cartograms. Görtler [40] proposed bubble treemap (Figure 2.9) that represents the uncertainty information using wavy circle boundaries, and varied wave frequency and amplitude-based on the uncertainty.

Nusrat et al. [78] examined how bivariate variables can be shown in cartograms (Figure 2.10). Since one variable is encoded as the area of a region, they mapped the other variable to a feature of the region boundary. Patterson and Lodha [82] encoded five socio-economic variables simultaneously on a world map using country fill color, glyph fill color, glyph size, country boundary color, and cartogram distortion.

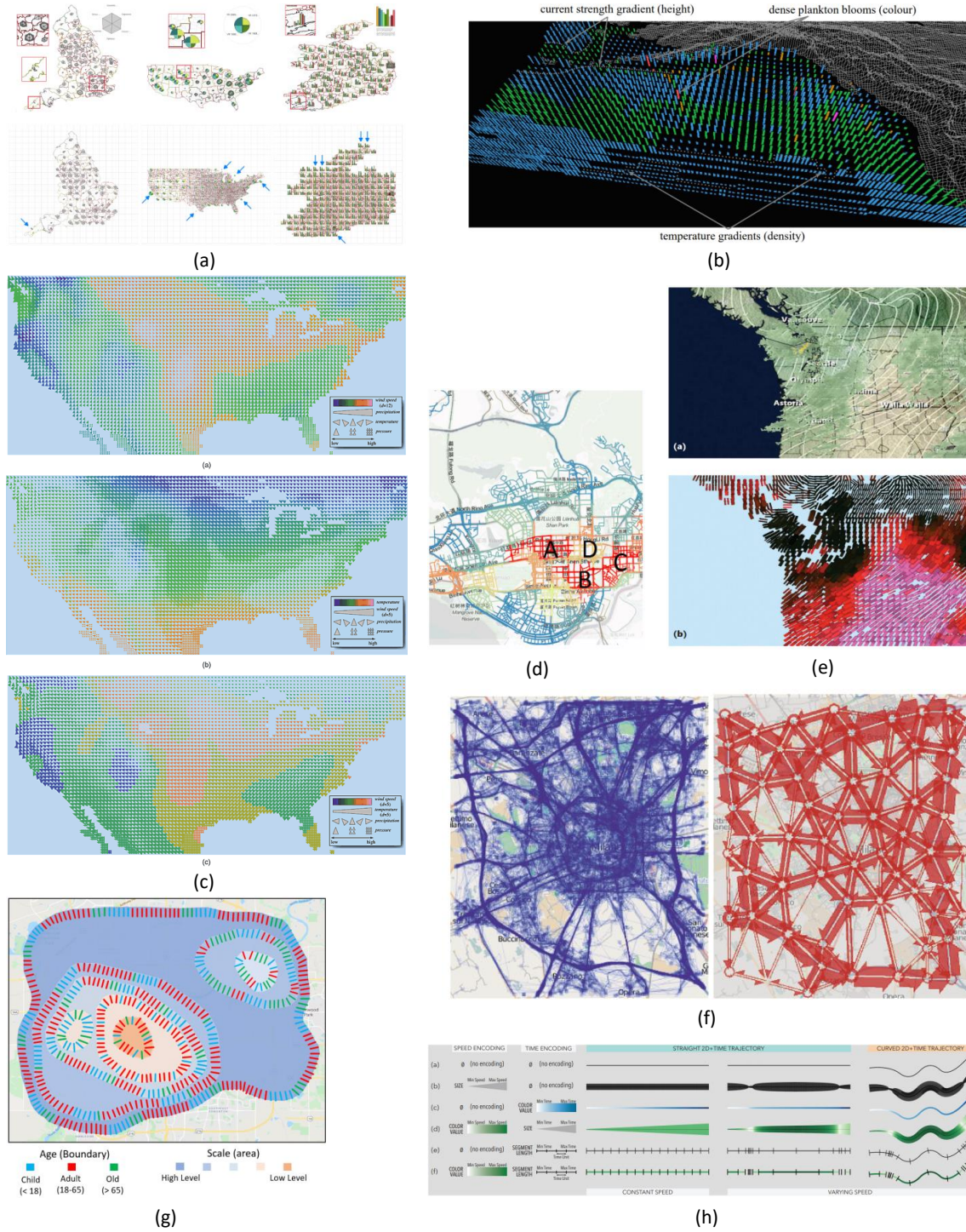


Figure 2.7: (a) Glyph-placement by the authors (top row) vs grid-based method (bottom row) [75] (b) visualizing open-ocean plankton density, current strength, and sea surface temperature with color, height, and density respectively [47] (c) a 4-variate map of US using color, orientation, size and density of a triangular glyph [46] (d) visualizing urban network centralities on a part of ShenZhen, China [55] (e) traditional Visualization (top) vs simulated brush strokes (bottom) varying color and texture [48] (f) the original 6,187 car trajectories (left) vs aggregated trajectories (right) [1] (g) visualizing age data over a region at multiple levels of spatial aggregation [106] (h) visualizing time and speed [83]

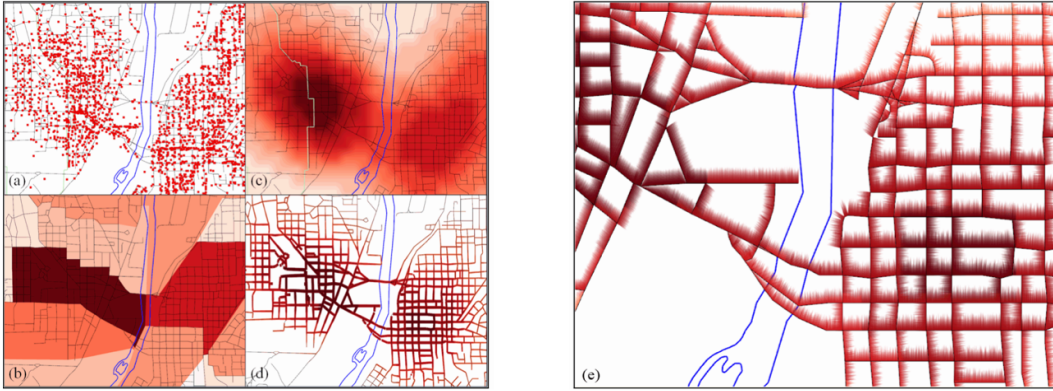


Figure 2.8: Visualizing crimes in West Lafayette and Lafayette, Indiana, where the blue line represents the Wabash river: (a) each point is a crime event (b) aggregating all the points by areal units (c) continuous domain approximation from point sampling (d) continuous domain approximation using solid lines on to the roads (e) authors' abstraction using a series of bristle lines applied along the roads [59]

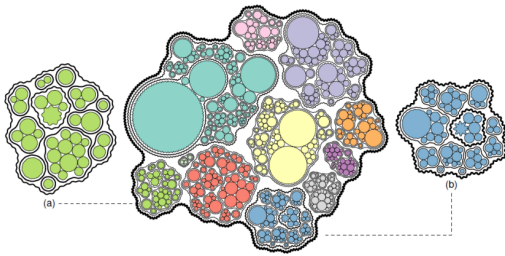


Figure 2.9: Showing a bubble treemap of the S&P 500 index, decomposed into sectors and companies. Each circle represents a stock, its area \propto the mean closing price, whereas the standard deviation is depicted using the out-lines [40]

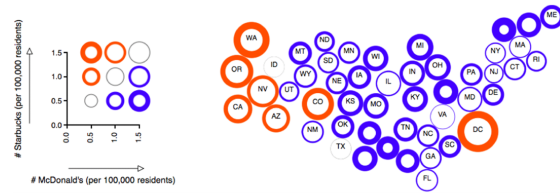


Figure 2.10: Visualizing the distribution of Starbucks and McDonald's shops per 100,000 residents in the US with a per-capita bivariate cartogram [78]

CHAPTER 3

DATASET DESCRIPTION

Visualization greatly depends on data distribution. Having a versatile data distribution is important to see how the visualization portrays multivariate data at different combinations at different places over the map. To ensure this, we used 2 main types of data in the studies- synthetic and real-life meteorological data.

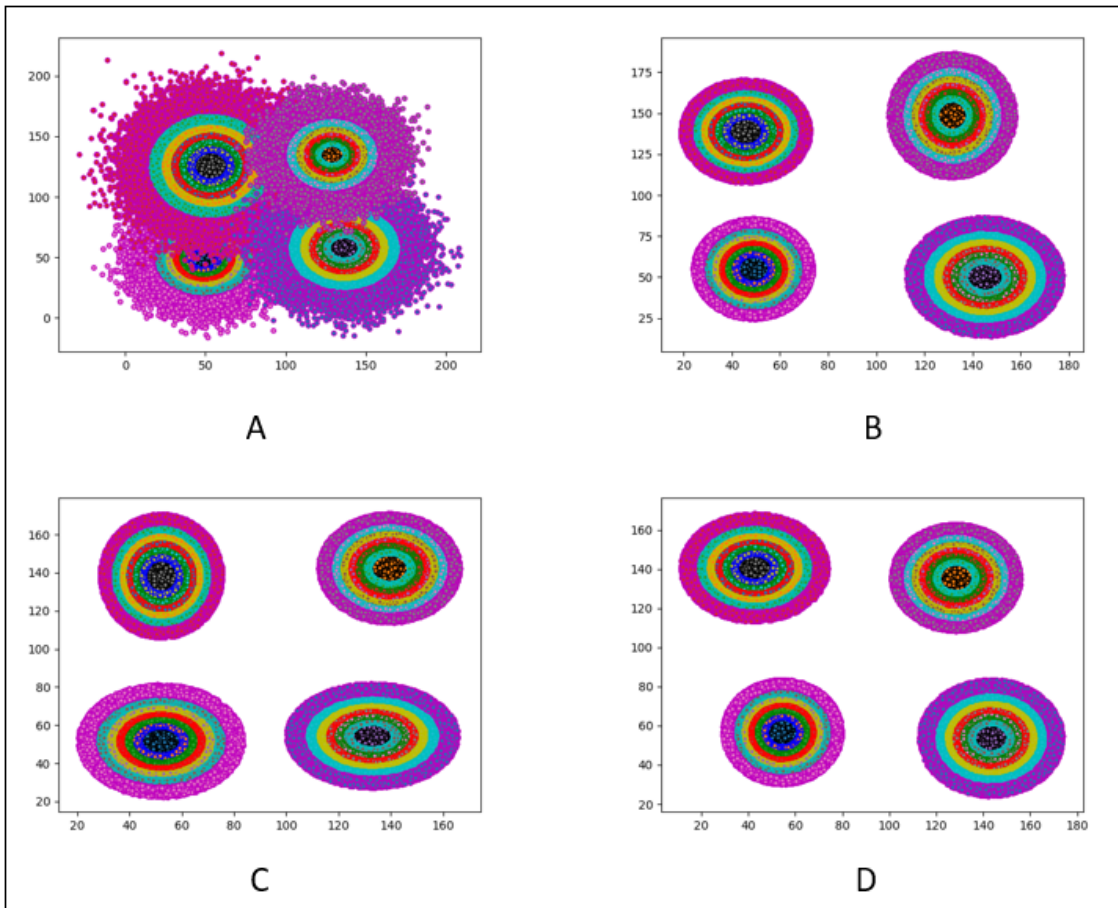
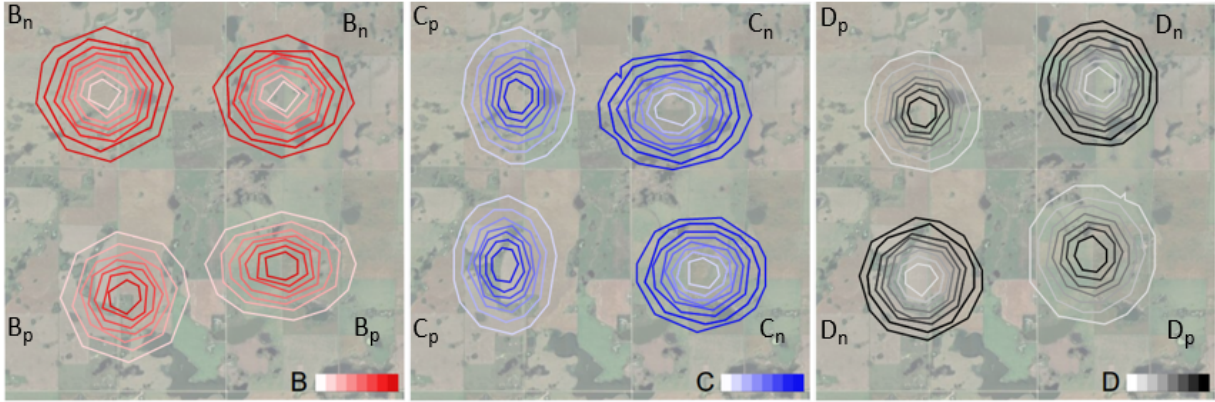


Figure 3.1: Synthetic data - A, B, C, and D scatter plots. Each set of concentric circles denotes a cluster. Each colored band indicates a contour band.

Dataset 1 – B, C, D



Dataset 2 – B, C, D

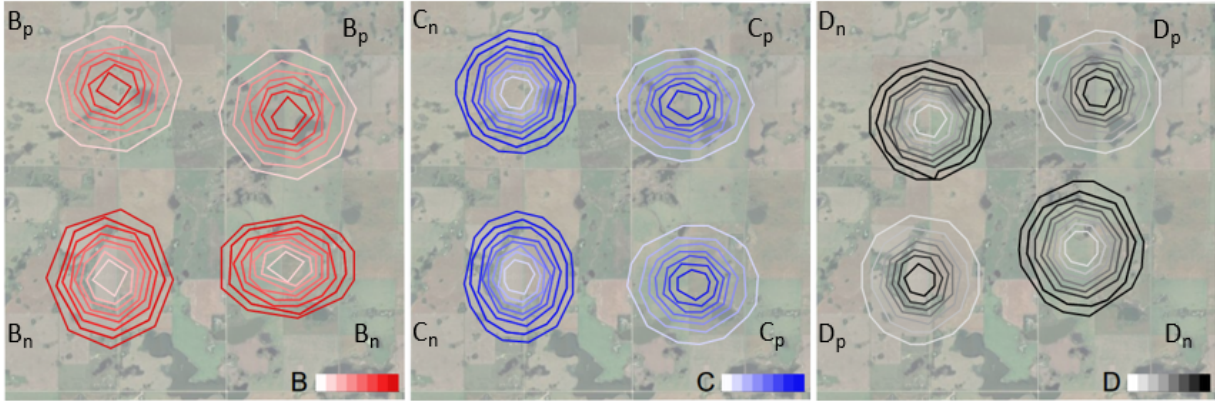


Figure 3.2: Synthetic data - B, C, and D contours with background. The shades indicate the values of contour bands- darker (lighter) shade means higher (lower) value.

3.1 Synthetic Data

3.1.1 Rationale

Real-life geospatial data variables can change frequently over the map, maintaining a trend, a pattern, or even exhibiting sharp changes due to some geospatial conditions. Common geospatial data tasks involve looking for a specific combination of data variables and taking decisions based on the findings. In order to make synthetic data that will mimic such a large variance in data distribution, we designed 6 different datasets for our study. While designing, our aim was to develop datasets that would have complex variable correlations at multiple points, which would reveal the strength and weakness of our visualizations at a more granular level.

3.1.2 Designing

The classic volcano dataset in Figure 1.1 shows the value increasing into the middle of the map from its borders. We denote such a change of a variable X as X_p , i.e. a positive change of X . Similarly, the opposite change can be denoted as X_n , where the values would decrease from the borders to the middle. We assume that each variable can have one of these two properties in a contour cluster (for B , it can be B_p or B_n). A cluster here is a set of concentric circles as shown in Figure 3.1. Therefore, at any cluster, there can be 8 different combinations of B , C , and D ($B_x C_x D_x$, where 'x' can be 'p' or 'n').

To visualize 3 variables (B , C , and D) with all 8 possible combinations, we created 2 datasets. Each dataset displayed 4 combinations in them in 4 clusters, positioned at 4 corners- top left, top right, bottom left and bottom right (Figure 3.1,3.2). The combination of $B_p C_p D_p$ means that, the values of all three variables increase from outside to inside. As depicted in Figure 3.2, the value changes are shown using color shades- darker shade means higher value and vice versa. For example, $B_n C_n D_n$, can be found at the top right corner of dataset 1 in Figure 3.2, where all variables' values decrease from outside to inside.

The value change of A has been kept fixed (A_p) at all clusters, i.e. the values increase inside. All the clusters of A , B , C , and D at a corner overlapped one another, making a specific data combination (Figure 3.3). This way, each dataset visualized 4 different combinations of A , B , C and D in 4 clusters.

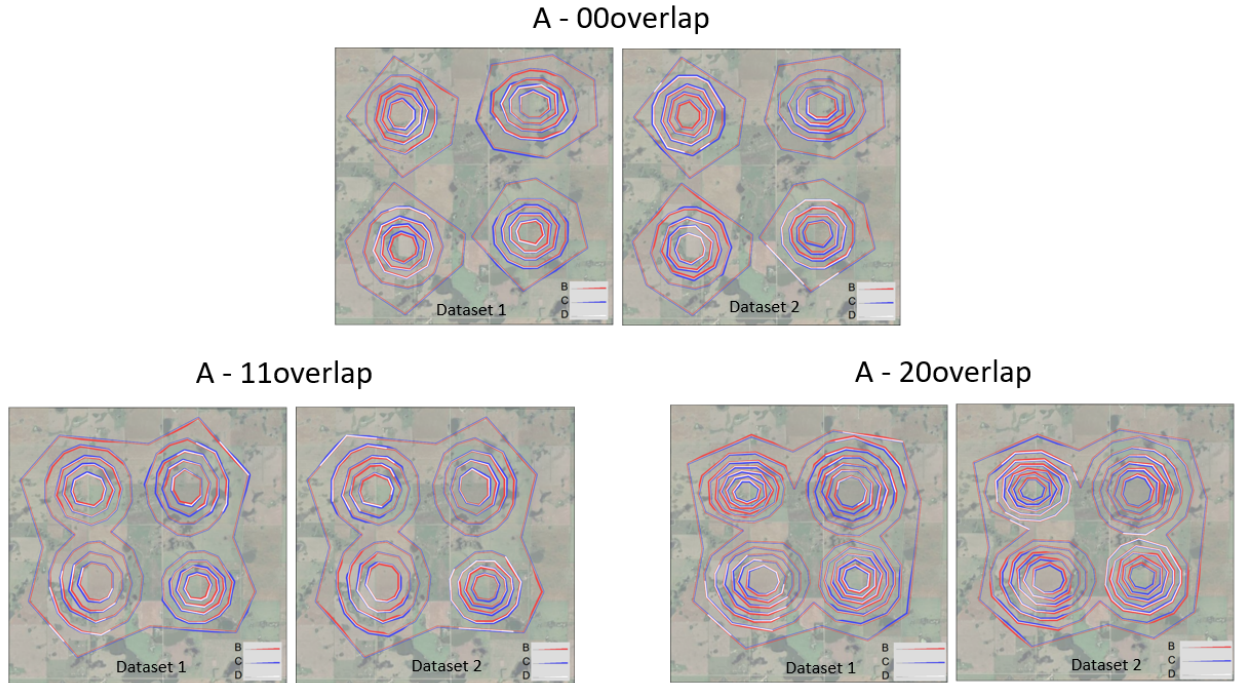


Figure 3.3: Synthetic data - final output

The clusters have their own centers, if the centers are far, their corresponding contour lines will not overlap. If they are close, the lines might overlap to some degree, depending on how close they are. Thus, the contour lines, as well as

the visualization will vary according to the degree of overlap.

We incorporated this idea of cluster merging into A's contour lines and created 3 different versions. We changed only A's cluster center positions, not others, as all the data were to be visualized on A's contour lines. We named them as *00overlap*, *12overlap* and *20overlap* (see Figure 3.3). *00overlap* has no overlap among clusters, *12overlap* has 1 contour line around all clusters with 2 left clusters merging inside and *20overlap* has 2 contour lines around all clusters with no cluster merging inside with one another. Therefore, we created 6 different datasets for the user study.

The clusters were created using python's `make_gaussian_quantiles()` function. Each cluster had 40000 samples, with varied covariance (2.5-7.5), 2 spatial features (x and y coordinate), and 4, 6, or 8 classes (4, 6, or 8 concentric circles to represent 4, 6, or 8 contour intervals respectively). The covariances of each variable's clusters were varied so that the clusters in the same dataset had differences in sparsity.

The clusters were then interpolated and reshaped into position using python's `np_interp()` and `.reshape()` functions respectively, into a 200×200 matrix, respectively, such that the point density plot for each cluster takes the shape of a peak or valley. While overlapping the A, B, C, and D clusters at a specific corner, e.g. top left, the centers of the clusters were kept closer but not at the same point, to ensure better visualization.

After the contours are generated, the outermost contour had a lot of cringes around it while the others also were a bit wavy (Figure 6.2 (left)). There were numerous small contours scattered in the center and along the border of the outer contour. We simplified the contours with a javascript function `.getSimplifiedContours()`. The function works based on a simplification factor, the higher the value, the aggressive the simplification is. We simplified the outer (inner) contour with a higher (lower) value. Thus the final contours were returned (Figure 6.2 (right)).

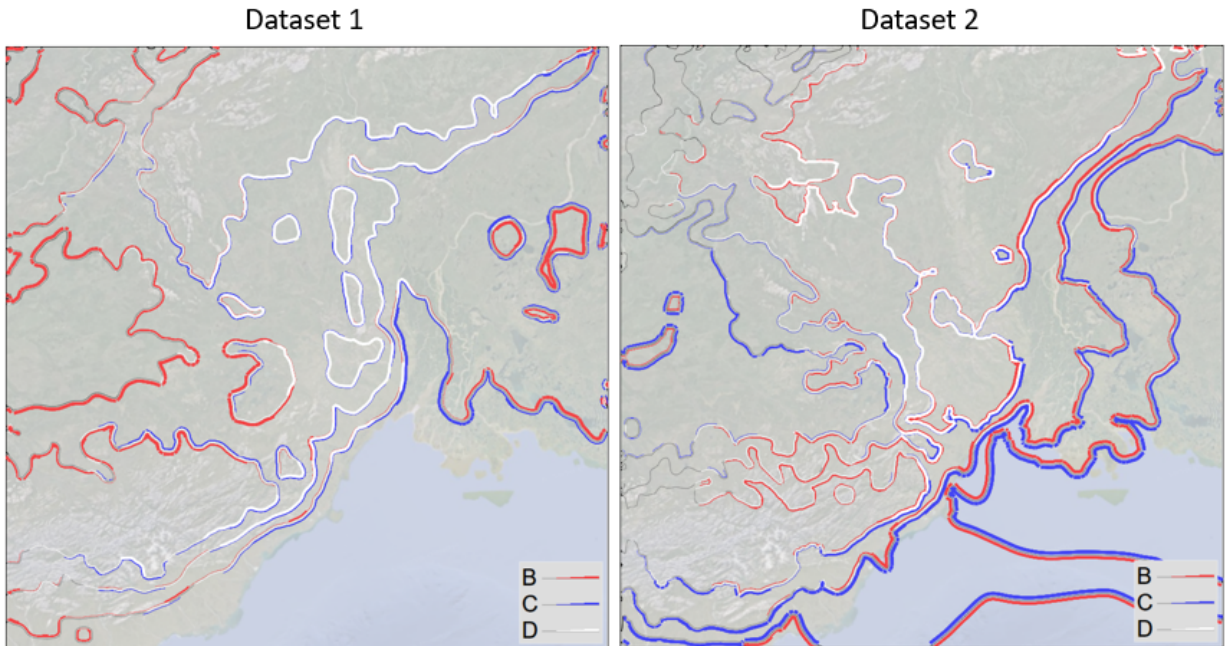


Figure 3.4: Real-life meteorological data overview

3.2 Real-Life Meteorological Data

To test the designs on real-life data, we used meteorological data of western Canada from the Weather Research & Forecast (WRF) dataset, which is generally kept in cloud storage called Graham Compute Canada cluster. The available amount of data ranges from the year 2008 till 2015 (8 years). Each data file consists of hourly (24) data for a single day over 699 latitude and 639 longitude points of western Canada. Each data file is of (approx.) 1.3 GB that contains 10 million (10,719,864) samples, each sample having around 36 weather parameters. The files come in .netcdf format.

For our study, we extracted 4 variables from the dataset of September 1, 2015 of timestamp 0 (00:00:00) over 499-698 latitude and 0-199 longitude using python. The 4 variables were- *Temperature*, *Pressure*, *Soil Moisture* and *Albedo*. The data were then filtered (25th and 75th percentile) and normalized. After normalization, we passed the data through a 5 by 5 mean filter to get a smoother image. Then the variables were shuffled to get 2 different datasets for the Study 3 (Figure 3.4).

3.2.1 Differences between Data Types

Two significant differences between synthetic and real-life data are the density and the position of contour lines. We controlled the density distribution (number of contour lines) using three different contour intervals in our synthetic data where 6 contour intervals (Figure 3.3) had 5 contour lines in each cluster. In a real-life scenario, the contour density is expected to vary frequently. While our synthetic dataset would have all possible variable combinations (increasing, decreasing) among three variables, some combinations may never in a real-life dataset.

We used synthetic data in the first 2 studies and added both synthetic and real-life data in Study 3. Adding both types in the first 2 studies would double the tasks and hence might result in a fatigue effect. Moreover, we will show in Study 3 that the results of synthetic data also generalize for real-life data, which establishes the validity of using only synthetic data in the first 2 studies without losing any potential finding.

CHAPTER 4

VISUAL ENCODING

In this section we describe five designs (Figures 4.1–4.5) for encoding geospatial information with four-attributes: A, B, C, and D. We assume that all the attributes are numeric and positive. We create a set of contour lines using the attribute A, and then encode the attributes B, C, and D along the contour lines of A using visual features. For encoding, we used visual features like line thickness, monochromatic color scheme, and pie slice. While combining pairs of features, we chose either blending (due to its common use in correlation analysis in geospatial data [35, 53, 89]), or thickness and monochromatic color scheme (following prior research [83]).

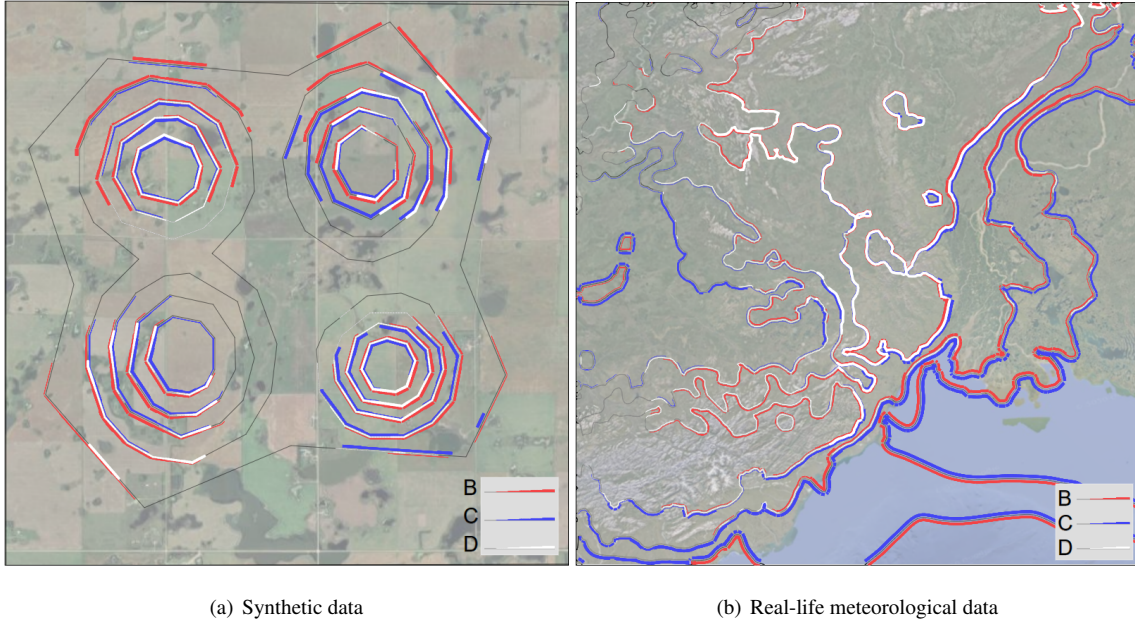


Figure 4.1: Visualizations using *Parallel Lines*

A contour line over a map is a polygon that consists of the point coordinates, where two consecutive points represent a line segment of the contour. Depending on the data distribution, a line segment can be very small or large. To get a fine detail of the data distribution of B, C and D on a contour line, each two consecutive point coordinates, i.e. line segment has been extracted and divided into a number of new points. The longer the line segment, the more it has been cut into new points. Then the values of B, C and D on those points have been calculated from their respective contour maps and visualized using different encoding schemes.

At any point over A's lines, the values of B, C, or D is the corresponding variable's contour band's threshold value. For example, if the value of B at any point is 0.30, then the point is in the interval of 0.25-0.50, for a 4 contour interval model (0, 0.25, 0.5, 0.75). Thus, the value returned will be 0.25, which is the threshold. The reason behind keeping the value as it's threshold is to minimize data variability, keeping the values discrete and making it easy to compare values.

Table 4.1: Visual encoding table: *Parallel Lines*

Variable	Encoding	Position	Color	Data Representation		Description
				Min	Max	
A	-	-	-	-	-	The contour lines of A
B	Line	At one side of D	#f44345 (Red)	Line segment width Zero	Line segment width Max	Wider line segment means higher B and vice versa
C	Line	At the other side of D	#4345f4 (Blue)	Line segment width Zero	Line segment width Max	Wider line segment means higher C and vice versa
D	Line	On A's contour lines, between B and C	#ffffff (White)	Line segment width Zero	Line segment width Max	Wider line segment means higher D and vice versa
B and C grow on either sides of D						

4.1 Design 1: *Parallel Lines*

This design maps B, C, and D into three parallel lines colored with distinct colors (Figure 4.1). The encoding is explained in detail in Table 4.1. The lines for B and C lie on opposite sides of the contour lines of A, and the lines for D overlap the contour lines of A. The data values are encoded using linewidth. Each line can have a minimum and maximum linewidths of 0 and w , respectively. The value of an attribute has been linearly mapped to the range $[0, w]$. If the value of D is 0, then the black contour line of A becomes visible. The legend illustrates B, C, and D by varying the linewidth.

The lines for B, C, D do not overlap at any point. Even if D is 0, one can see a constant distance between B and C lines, which makes it easy to interpret the individual variables and also compare them.

The reason behind drawing the black thin lines of A is to give the feeling of continuity at points where there is no B, C or D.

Table 4.2: Visual encoding table: *Color Blending*

Variable	Encoding	Position	Color	Data Representation		Description
				Min	Max	
A	-	-	-	-	-	The contour lines of A
B	Color	Inside D's lines	Shades from	Color	Color	Darker Shade means higher B and White means Zero B
			White to #f44345 (Red)	White	#f44345 (Red)	
C	Color	Inside D's lines	Shades from	Color	Color	Darker Shade means higher C and White means Zero C
			White to #4345f4 (Blue)	White	#4345f4 (Blue)	
D	Line	On A's contour lines	-	Line segment width Min	Line segment width Max	Wider line segment means higher D and vice versa
B and C are color-blended on A's contour lines having width proportional to D						

4.2 Design 2: *Color Blending*

This design encodes B and C with distinct colors and then blends them on the contour lines of A (Figure 4.2). The encoding is explained in detail in Table 4.2. The attribute D is mapped to the width of the contour line. Note that since the contour line of A has a non-zero width u , the values of D are mapped to the linewidth range $[u, w]$. Consequently, B and C remain visible even when D is 0. The legend illustrates B and C using a blending matrix, and D using a line of increasing width.

The number of colors in the scale depends on the number of contour intervals. The synthetic data (left) visualization of Figure 4.2 has 6 contour intervals and 6 color shades for both B and C in the legend. The real-life (right) data has 4 intervals, hence 4 color shades.

The values of B and C at any point is mapped to the colors from their respective discrete sequential color scales. The 2 colors are then blended using the formula of '*multiply mix-blend-mode*' of CSS where the corresponding r, g, and b components of C and D are multiplied to get the final r, g, and b components of the visualization. For example, when the value of B is 0 and C is the maximum, that line segment color becomes the maximum shade of C from C's color scale. The opposite happens when B is maximum and C is 0, i.e. line segment color becomes the maximum shade of B from B's color scale. When B and C are equal, the color comes from the color legend's diagonal.

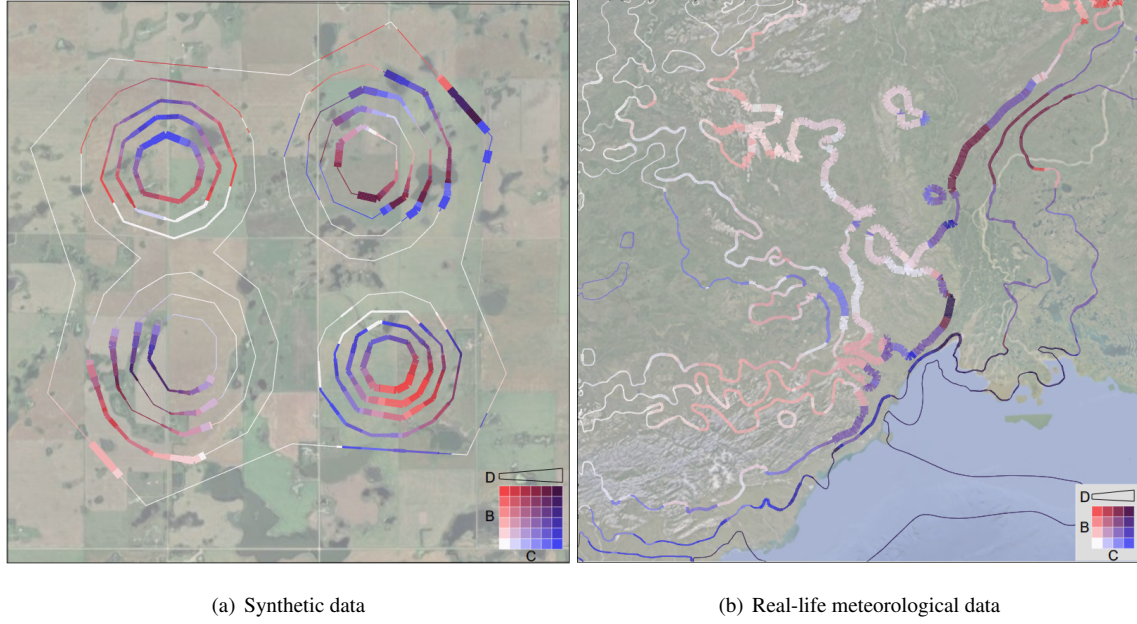


Figure 4.2: Visualizations using *Color Blending*

4.3 Design 3: *Pie*

This design encodes B, C, and D using pie slices of distinct colors, and put them together to create a pie icon (Figure 4.3). The encoding is explained in detail in Table 4.3. The only difference from a pie chart is that the sum of the values of B, C, and D may not be equal to the total pie area. The pie icons are placed successively along the contour line of A. The pie slices for B, C, D starts at 0° , 120° and 240° , respectively (assuming the top mid as 0°), and can grow clockwise to cover an angle of 120° . An attribute value is encoded into the angle covered by the corresponding pie slice. The legend illustrates the attributes using a pie icon.

The amount of area to fill is proportional to the variable's value. Whatever the value is, it gets multiplied by 120° to get the central angle for filling up the area. For example, if D has a value of 0.5, the pie area from central angle 240° to 300° (120×0.5 is 60) is filled with D's color. If a variable value is 0, that variable's allotted area in the pie remains empty.

4.4 Design 4: *Thickness-Shade*

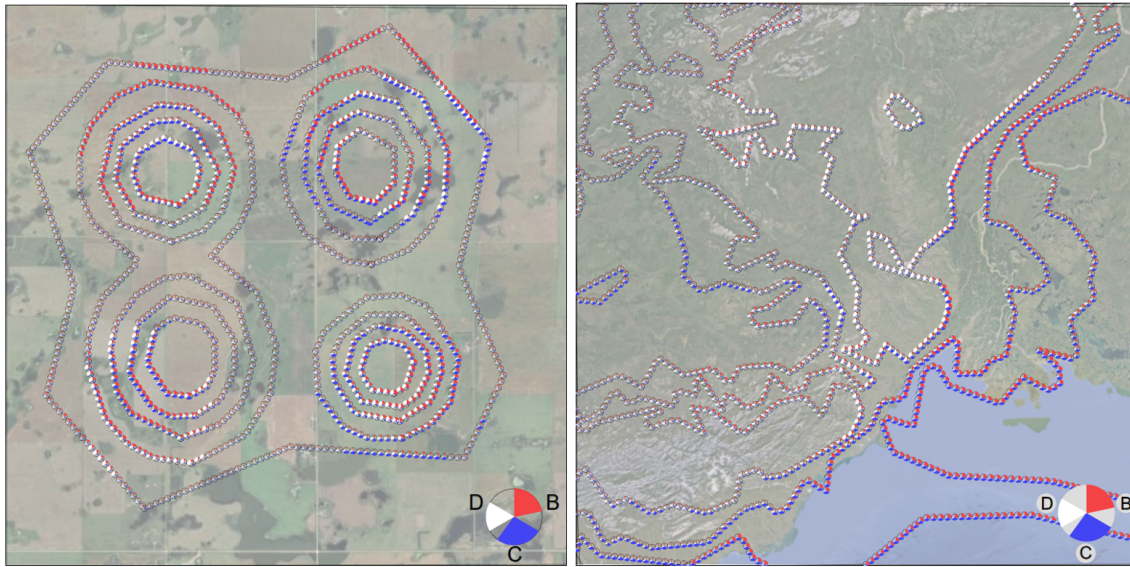
This design represents B and D using two distinct lines (Figure 4.4). The encoding is explained in detail in Table 4.4. The lines of B and D lie on opposite sides of the contour lines of A. Both the values of B and D are encoded using linewidth. The values of C are encoded using a monochromatic color scheme, where the color appears on B's line. A low (high) C value corresponds to a lighter (darker) shade. The minimum linewidth for B is set to a positive threshold u , making a range of $[u, w]$ so that C remains visible even when B is 0. The legend illustrates B, and D using lines of varying width, and C using its color scheme.

Table 4.3: Visual encoding table: *Pie*

Variable	Encoding	Position	Color	Data Representation		Description
				Min	Max	
A	-	-	-	-	-	The contour lines of A
B	A piece of the Pie	Starts at 0°(top) of the pie	#f44345 (Red)	Empty area from 0° to 120°	Area from 0° to 120° filled with #f44345	Higher area covered means higher B and vice versa
C	A piece of the Pie	Starts at 120° (clockwise from top) of the pie	#4345f4 (Blue)	Empty area from 120° to 240°	Area from 120° to 240° filled with #4345f4	Higher area covered means higher C and vice versa
D	A piece of the Pie	Starts at 240° (clockwise from top) of the pie	#ffffff (White)	Empty area from 240° to 360°	Area from 240° to 360° filled with #ffffff	Higher area covered means higher D and vice versa
B, C and D each has 1/3 of the whole pie, starting from fixed positions						

4.5 Design 5: *Side-by-Side*

This design shows B, C, and D in separate side-by-side views (Figure 4.5). The encoding is explained in detail in Table 4.5. Each of B, C, and D is encoded using a distinct monochromatic color scheme. The color appears on the contour lines of A. The legend at each view illustrates the color scheme used to encode the corresponding variable. During our experiments, we ensured that the number of pixels used for *Side-by-Side* is comparable to those of the other designs. In particular, we choose the width and height of each of the *Side-by-Side* view to be $\lceil \sqrt{PA/3} \rceil$, where PA is the total pixel area of any other design, assuming a square display.



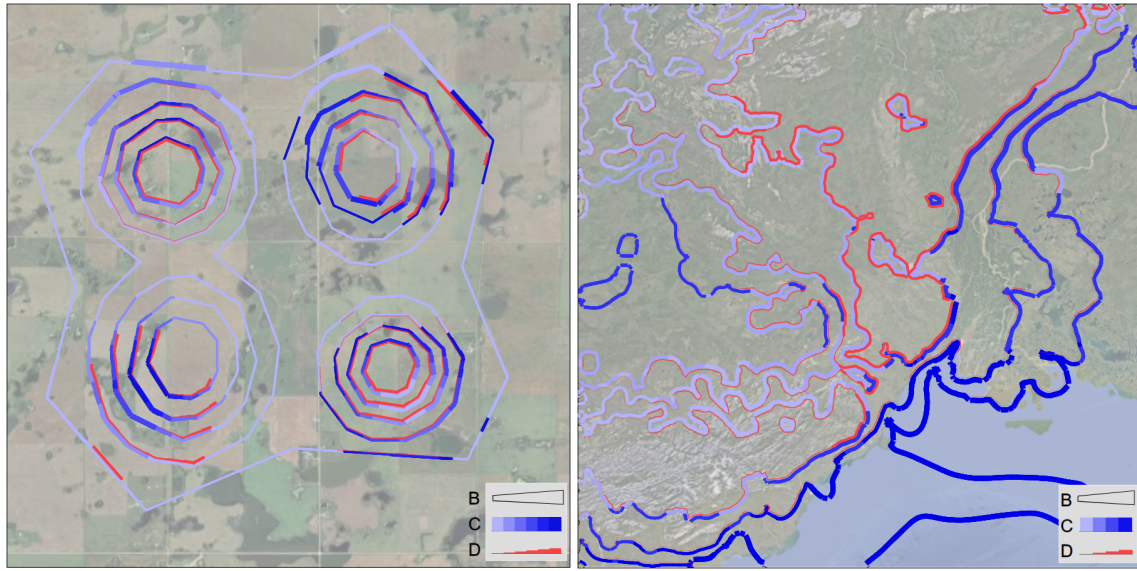
(a) Synthetic data

(b) Real-life meteorological data

Figure 4.3: Visualizations using *Pie*

Table 4.4: Visual encoding table: *Thickness-Shade*

Variable	Encoding	Position	Color	Data Representation		Description
				Min	Max	
A	-	-	-	-	-	The contour lines of A
B	Line	At one side of A's contour lines	-	Line segment width Min	Line segment width Max	Wider line segment means higher B and vice versa
C	Color	Inside B's lines	Shades from White to #4345f4 (Blue)	Color White	Color #4345f4 (Blue)	Darker Shade means higher C and White means Zero C
D	Line	At the other side of A's contour lines	#f44345 (Red)	Line segment width Zero	Line segment width Max	Wider line segment means higher D and vice versa
B changes the width of the line that has C in it, C changes shade of color inside the B, line D changes the thickness of the red line adjacent to B						



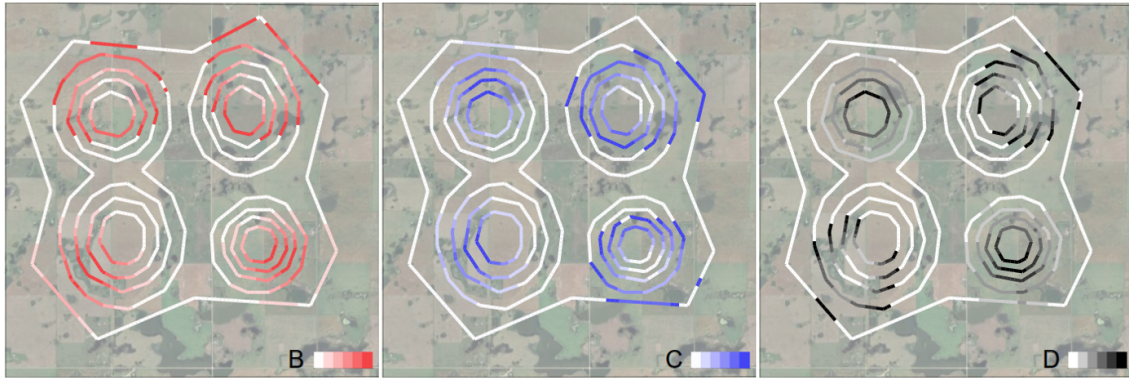
(a) Synthetic data

(b) Real-life meteorological data

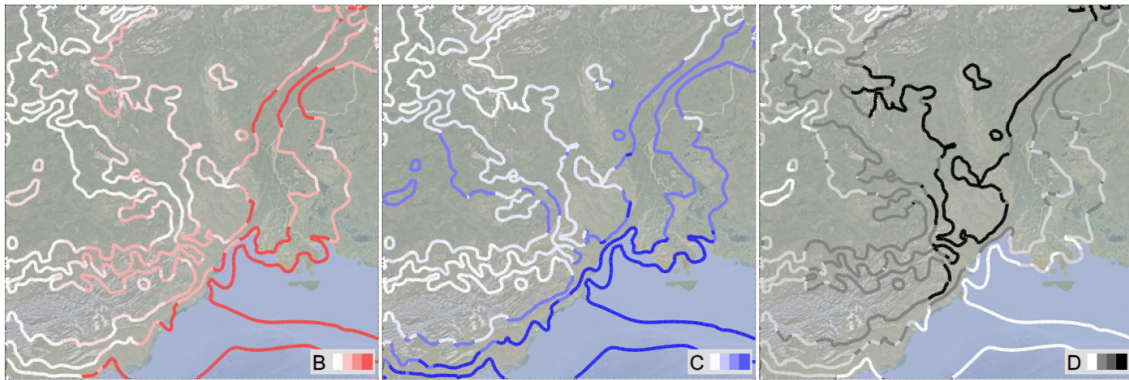
Figure 4.4: Visualizations using *Thickness-Shade*

Table 4.5: Visual encoding table: *Side-by-Side*

Variable	Encoding	Position	Color	Data Representation		Description
				Min	Max	
A	-	-	-	-	-	The contour lines of A
B	Color inside A's contour lines	Left viewport	Shades from White to #f44345 (Red)	Color White	Color #f44345 (Red)	Darker Shade means higher B and White means Zero B
C	Color inside A's contour lines	Middle viewport	Shades from White to #4345f4 (Blue)	Color White	Color #4345f4 (Blue)	Darker Shade means higher C and White means Zero C
D	Color inside A's contour lines	Right viewport	Shades from White to Black	Color White	Color Black	Darker Shade means higher D and White means Zero C
B, C and D are visualized in separate views						



(a) Synthetic data



(b) Real-life meteorological data

Figure 4.5: Visualizations using *Side-by-Side*

CHAPTER 5

REAL WORLD EXAMPLES

This section demonstrates few real-world applications of the proposed designs to show their effectiveness in visualizing the underlying data distribution for detecting trends, correlations, anomalies, similarities, and/or dissimilarities.

5.1 Detecting Variable Values

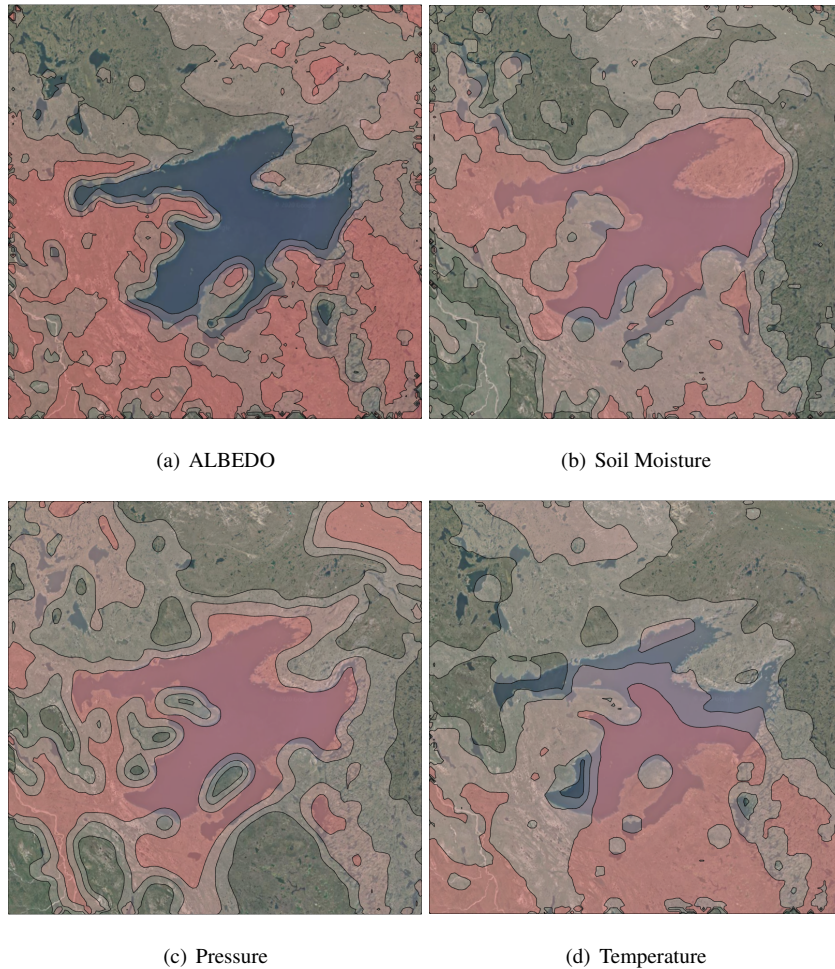
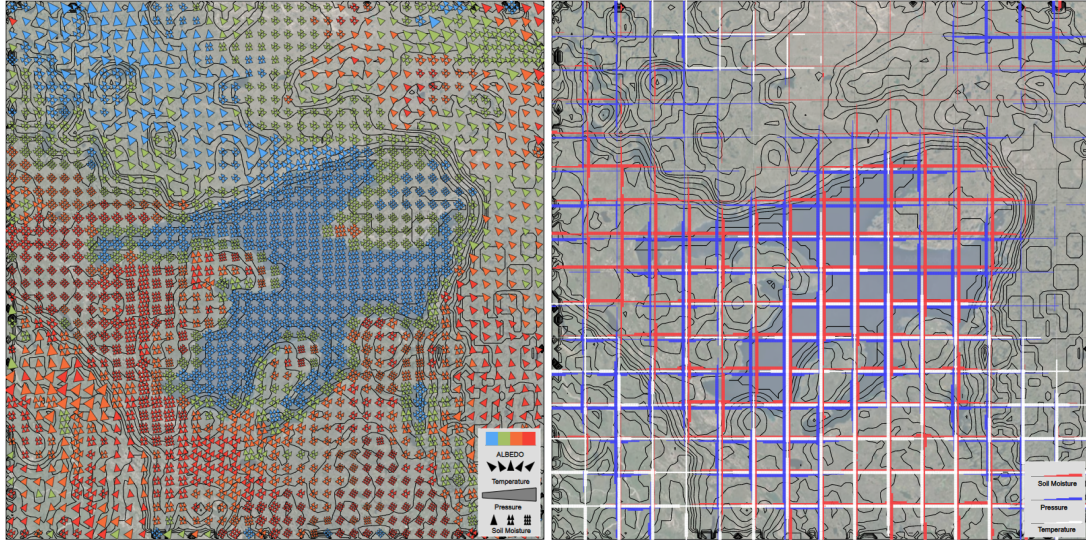


Figure 5.1: Distribution of 4 variables over the Great Bear lake of Canada (Date: April 1, 2015 00:00AM). The contour bands are color filled according to values from no color (low) to darker shades of red (high).



(a) Glyph visualization

(b) Grid based visualization

Figure 5.2: Multivariate visualization with (left) glyphs occludes the map, and (right) grid stylization lacks the gradient information

Figure 5.1 shows the distribution of 4 variables (ALBEDO, Soil Moisture, Pressure, and Temperature) over the Great Bear lake of Canada on April 1, 2015, using their respective contour lines in 4 different contour map views. The underlying map does not exactly overlap the top visualization layer as the background image is from 2020, where the data is from 2015.

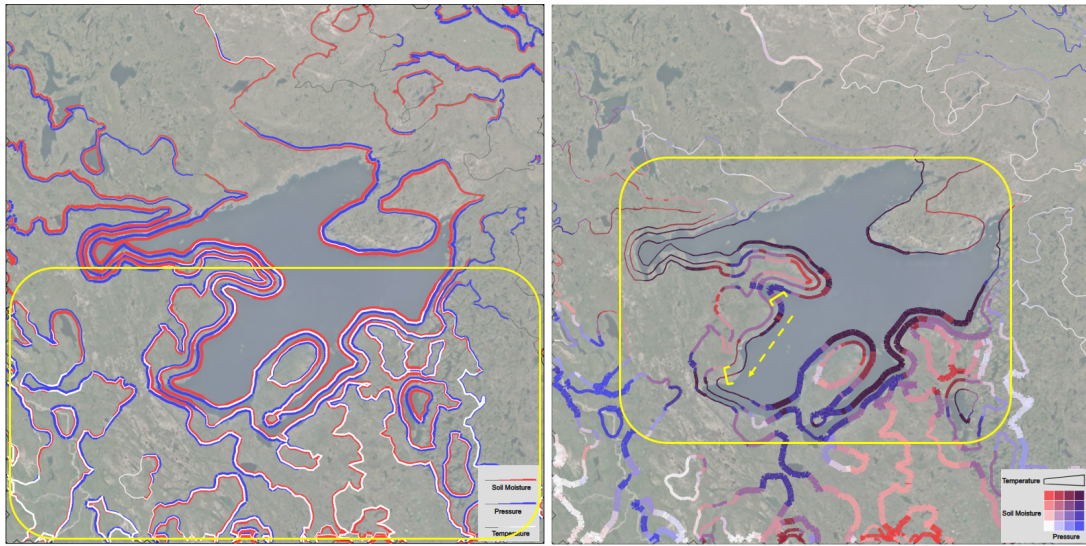
Looking at four different views to check data distribution, correlation, or trend is time-consuming, tiresome, and often ineffective. Alternatively, overlapping the four contour plots on top of each other in a single view, to compare all variable values at a specific point or check the trend of one with respect to another is not a feasible idea either.

Figure 5.2 shows two alternatives of visualizing multivariate data- using glyphs (left) or evenly spaced grids (right). Despite the popularity of glyphs conveying multivariate data, they suffer from the limitation of cluttering the underlying map [19]. For the grid-based visualization, the lack of geospatial connectedness and gradient information compromises the user performance.

Our designs (Figure 5.3) can show all four variables altogether, leveraging the contour lines of ALBEDO, by encoding Soil Moisture, Pressure, and Temperature on them. The visualizations reveal the relations among variables and also the trends along a direction. Trend can be seen along a contour line or across multiple contour lines. We discuss such a few examples of simple and insightful interpretations from each design.

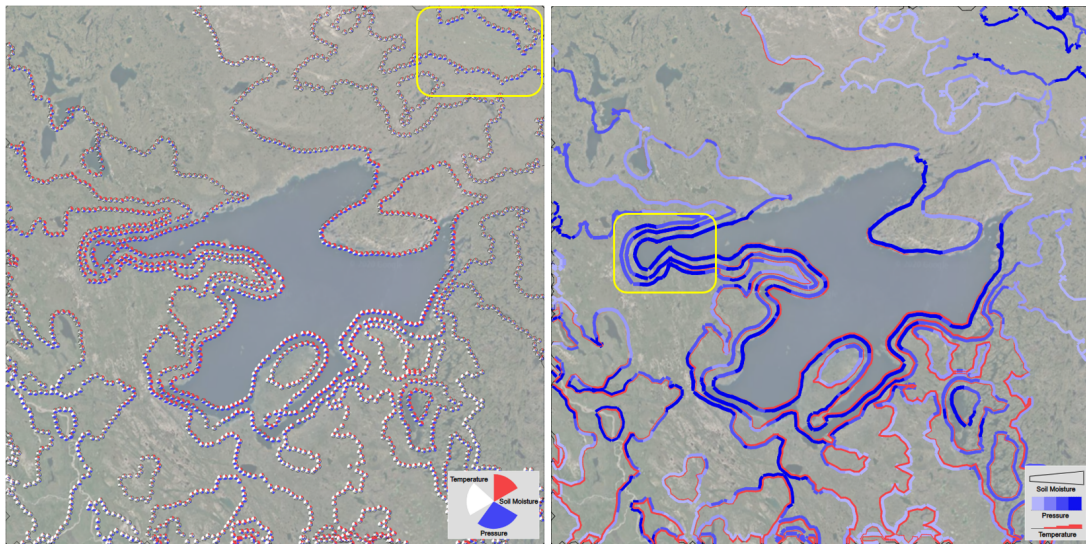
Based on the yellow markers from the designs:

1. Visualization through *Parallel Lines* shows that the temperature over the map is relatively higher at the bottom half of the map. We can see the wider white lines there, encoding temperature using line width.
2. From the *Color Blending* design we can see that, there are relatively higher soil moisture and higher pressure values along the border contour lines of the lake. Regardless of the width of the contour lines (line width encodes



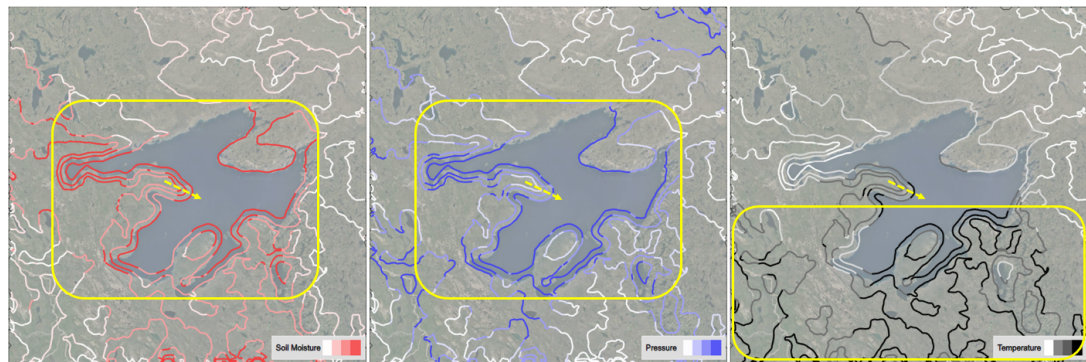
(a) *Parallel Lines*

(b) *Color Blending*



(c) *Pie*

(d) *Thickness-Shade*



(e) *Side-by-Side*

Figure 5.3: 5 different visualizations of the same data from figure 5.1

temperature), the darker shades of color denote higher soil moisture and pressure. Moreover, if we follow the marked contour part along the dashed line direction, we can see the decrease of temperature along the contour line.

3. The top right region of *Pie* shows higher pressure, where we can see more blue in the pies than its surrounding.
4. The marked region in the *Thickness-Shade* design denotes high soil moisture, high pressure, and no temperature, as we can dark blue line with high width, and absence of red.
5. The *Side-by-Side* design reveals higher soil moisture (red) and higher pressure (blue) at the lake borders and higher temperature (black) at the bottom half of the map. As a trend across multiple contour lines, the dashed line shows an increase in all 3 variable values along its direction.

It is possible to find more such trends, correlations, and interpret relative value differences among variables both along and across contour lines, which are shown in the later examples. Such interpretations can help practitioners gain important insights from the underlying data.

5.2 Trend Analysis

We present two trend analysis examples to demonstrate the trends of two variables (temperature, soil moisture). For each variable, data averages of 3 consecutive months were collected as B, C, and D respectively and visualized on contour lines, to find the inherent trend. The resulting visualizations show that similar and different relations can be easily detected through our technique.

5.2.1 Trend: Temperature

The weather over the US and Canada usually gets colder gradually from August to December, as winter approaches. We visualize the temperature trend in Figure 5.4, which shows the temperature data for the months of September (red), October (white), and November (blue) 2017 separately in the left column and our *Parallel Lines* design in the right. Those colored contour maps (left column: green-red) show that the temperature has decreased gradually over 3 months. We normalized each monthly data considering the maximum and minimum values among all three months combined. Contours lines were drawn based on September using four contour thresholds. Each monthly data (red, white, and blue lines) on the visualization had eight contour thresholds ([0, 0.125, 0.25, 0.375, 0.5, 0.625, 0.75, 0.875]). Each encoding of B, C, and D lines had maximum thickness of 4 pixels, making each contour line 0-12 pixels.

Rather than looking at three different contour maps for checking trends, one can get a clear picture of how the values are changing from our visualization. The red (September) lines have the highest width where the width decrease gradually in white (October) and then in blue (November) lines. The value change is more prominent at the top of the map where the values decreases the most. At the bottom, the difference reduces as there is less decrease in temperature than at the top.

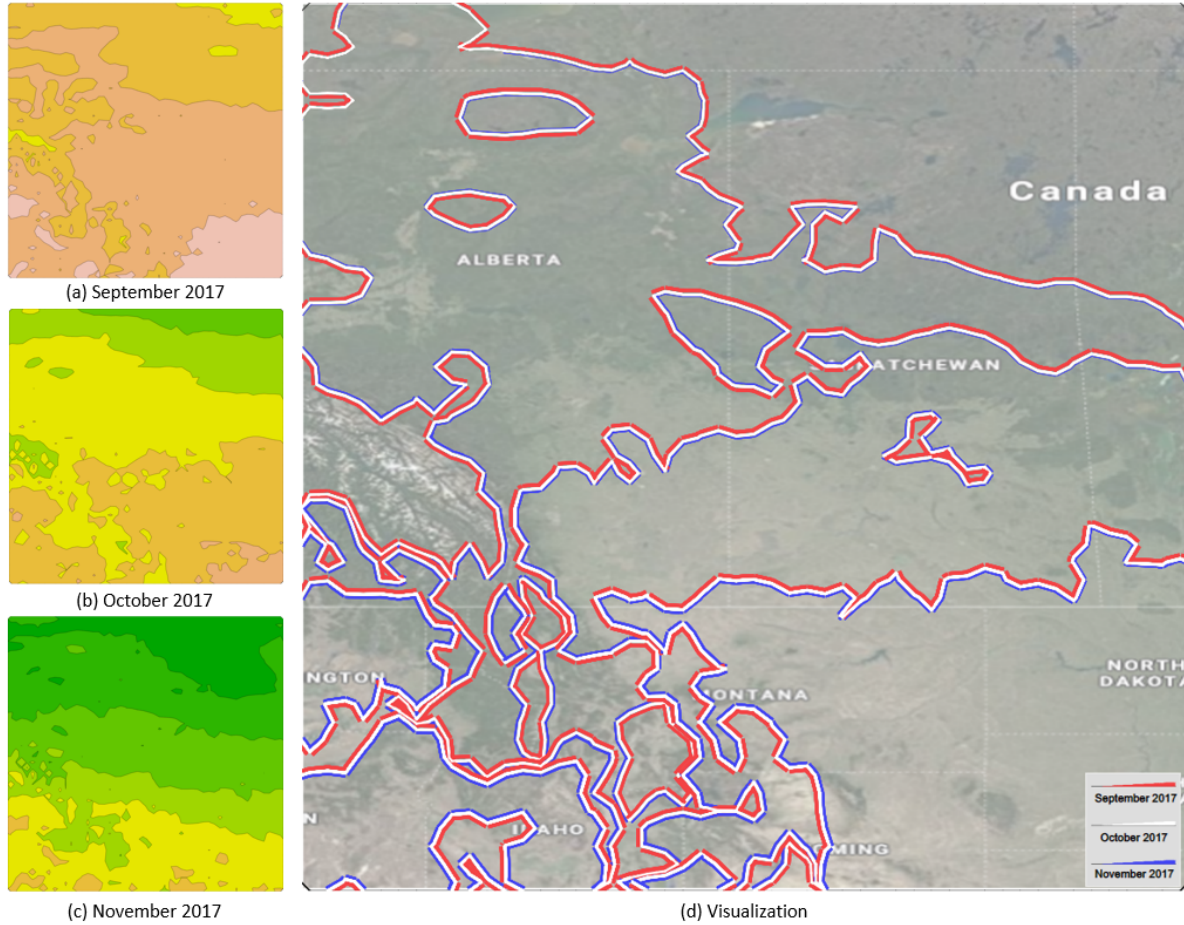


Figure 5.4: Trend: monthly average temperature- September, October, and November 2017

This example shows the effectiveness of our approach in analyzing trends with less effort and time, as the user does not have to look at three different maps to see the trend at a specific area of the map.

5.2.2 Trend: Soil Moisture

Figure 5.5 shows soil moisture data over the US and Canada for the months of May (red), June (white), and July (blue) 2020 separately in the top row. Those colored contour maps (green-red) show that the soil moisture at the bottom right and left corners have decreased over 3 months. The corresponding data have been visualized in the bottom row, Figure 5.5(d) is the same data, and Figure 5.5(e) is drawn using an exponential mapping of the data values. Similar to the previous temperature trend example, data was normalized considering the maximum and minimum values among all three months combined and each monthly data had eight contour thresholds. Contours lines were drawn based on May using those eight intervals. The encoding used the same maximum thickness of 4 pixels.

Figure 5.5(d) shows that the soil moisture at the two yellow marked regions at the bottom decreased from May to July, which can be seen in the separate top row images. Those two marked regions are part of the dry land of Arizona, which becomes drier as summer comes. The bottom right portion of the map covers Arkansas, Oklahoma, Kansas, and

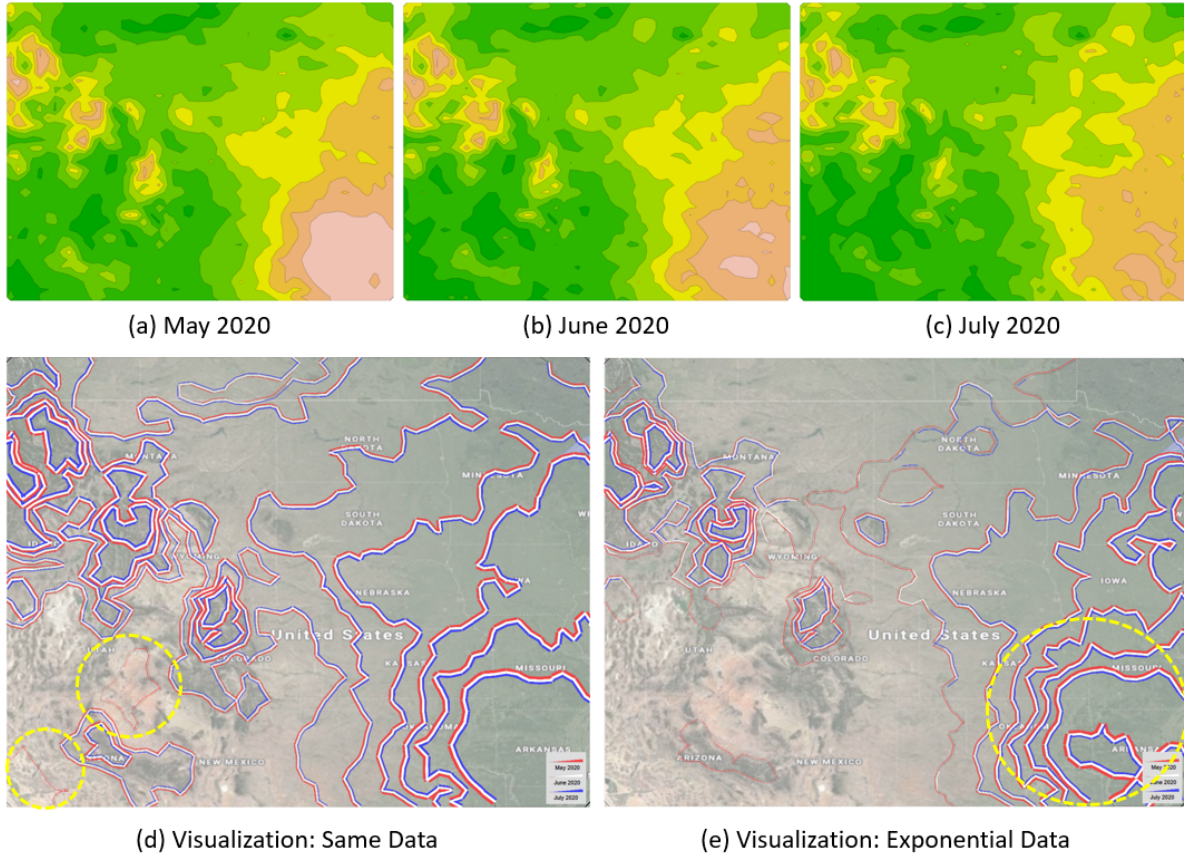


Figure 5.5: Trend: monthly average soil moisture- May, June, and July 2020

Missouri, where the value decreases by a small amount. That is why the difference can not be seen that clearly in Figure 5.5(d). After taking the exponential of the data, we could see noticeable difference in the yellow marked region of Figure 5.5(e).

This example leads us to make three observations. Firstly, contour interval and thickness have an important effect on visualization. Secondly, we used a visualization of 800×800 pixels on a laptop, where a bigger display should be able to help users see more subtle differences. Thirdly, given optimum display size and visualization parameters, our design has the potential to find a wide range of trends in multivariate geospatial data.

5.3 Front Visualization

When two air masses of different temperatures and, in most cases, also different moisture levels come in close contact with each other, the border drawn along their contact is called a front [97]. Weather fronts are very important in weather analysis and prediction, as they cause various pivotal weather conditions such as rain, snow, hail, thunderstorm, cyclone, and others. The development of a front depends on a few factors- temperature, moisture, wind direction, and pressure. Fronts usually develop close to low-pressure zones and move along the direction of the wind.

There are four types of fronts that can occur- cold, warm, stationary, and occluded fronts. Cold fronts occur when

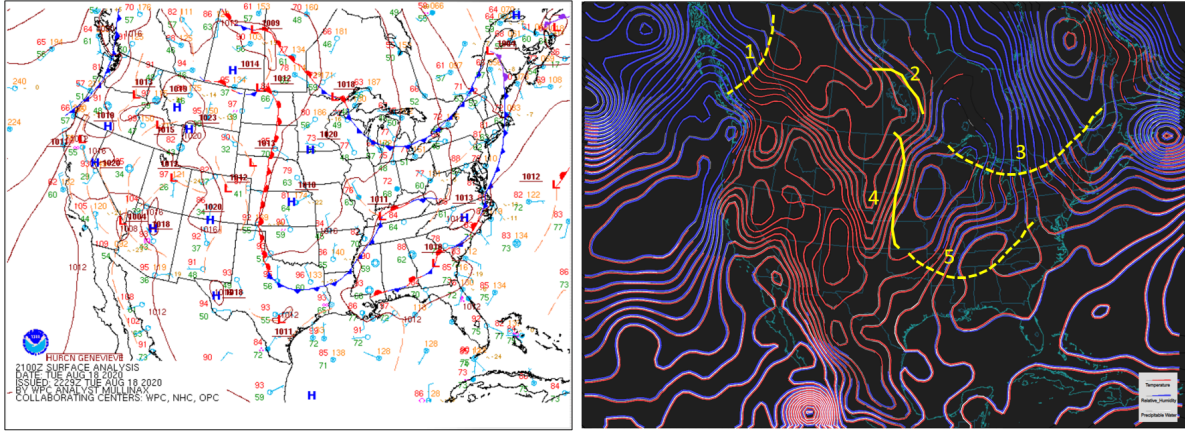


Figure 5.6: Front detection using *Parallel Lines* (right) and comparing with the actual NOAA data (left))

cold air mass pushes under warm air mass which can result in a thunderstorm due to high moisture in the air. Cold fronts usually occur in the west of surface low pressure having winds moving from west or northwest behind it and east or southeast wind in front. Warm front occurs when a warm air mass slides up over a cold air mass, developing having a broad area of clouds, resulting in a gentle rain or light snow. Warm fronts usually occur in the east of surface low pressure. The wind direction does not always follow a fixed pattern, for which wind is considered as a secondary indicator. But the things that certainly occurs along a front is a wind shift (change in wind direction). Stationary front occurs when both cold and warm air meet and neither of them can move each other, which can result in cloud and precipitation of several days. Cold fronts typically move faster than warm fronts. Sometimes a very strong cold front catches a warm front and another cold front, pushing both of them away, developing in an occluded front. This front can bring strong winds and heavy precipitation.

Fronts change position frequently, and it requires a continuous prediction of weather conditions due to the dynamic characteristics of weather variables. Figure 5.6 shows the effectiveness of our design in detecting fronts. The data were taken from [National Oceanic and Atmospheric Administration \(NOAA\) Weather Prediction Center \(WPC\) archive](#). We visualized 4 weather variables (pressure, temperature, relative humidity, and precipitable water) of August 18, 2020, 9 PM over the maps of the US using our *Parallel Lines* design. The design was then compared to its actual data to check whether the fronts can be detected in our visualization. The contours were generated based on pressure, where temperature, relative humidity, and precipitable water have been visualized as red, blue, and white lines respectively. All variables have been normalized before visualization.

Figure 5.6 (left) shows five marked curved lines drawn over the visualization. Dashed and solid lines are used to represent cold and warm fronts respectively. The red lines in the middle clearly differentiate the west part of the US land area from the ocean (left) where there is dry weather (high temperature, low relative humidity) due to the underlying topology.

We discuss the cold fronts first. Line 1 has a lower temperature and higher relative humidity on its left, which depicts cold air mass. On its right, there is a high temperature with zero relative humidity. The border clearly shows a cold front, as depicted in the NOAA visualization (Figure 5.6 (right)). Line 3 and 5 show similar criteria too. All three lines are

located west of low-pressure points (see the 'L' and 'H' notations in NOAA figure). For warm fronts, both lines 2 and 4 have a higher temperature and lower relative humidity on their left and they are located east of two low-pressure points.

There are more fronts in the NOAA images that can not be depicted from the visualization. This can have several reasons. Firstly, the NOAA images are drawn taking wind direction, and dew point in consideration, which is absent in our data. Secondly, fronts at the places with fewer contour lines are difficult to detect. If more information can be provided, it could be easier to detect all the fronts from our visualization.

CHAPTER 6

IMPLEMENTATION DETAILS

The designs have been implemented using SVG (Scalable Vector Graphics) with *D3.js* [8] (web visualization library). The implementation steps are described in the following.

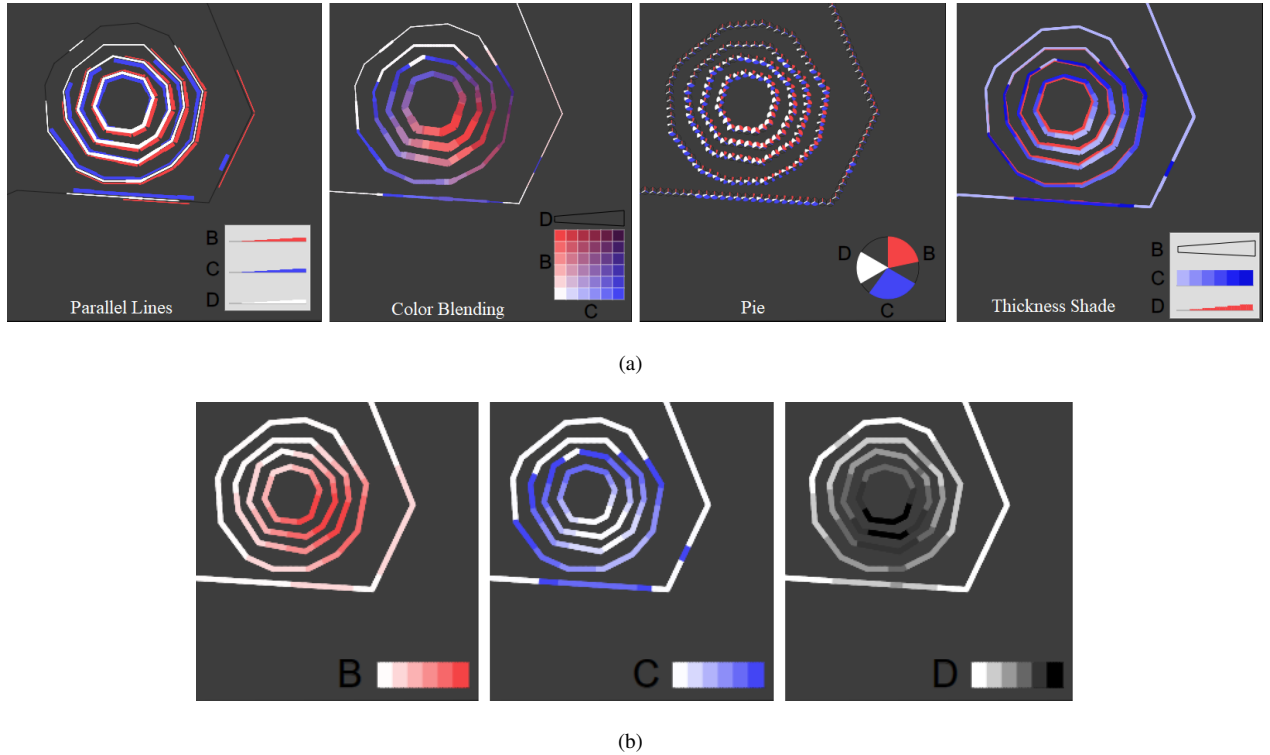


Figure 6.1: A closer look at the designs (the bottom right parts) with their respective legends. *Parallel Lines*, *Pie* and *Side-by-Side* have symmetric encoding scheme, unlike *Color Blending* and *Thickness-Shade*

6.1 Display Initialization

We used *d3.contours()* function to get the contour lines for A. If the number of contouring threshold is k , then we generated the threshold values using k -quantiles. The *d3.contours()* function returns an array of objects, where each object consists of a set of polylines representing a contouring threshold. We further processed these polylines by dividing long line segments uniformly to create fine-grained polygonal chains. We then encoded the attributes interpolating the

values at the endpoints of these tiny segments. For any endpoint, we retrieved the B, C, and D values by locating the point in the corresponding contour plot.

For simplicity, a square SVGs was used. In all designs except the *Side-by-Side* one, an SVG of 800×800 pixels was used, making a total of 640,000 pixels. In *Side-by-Side*, 3 separate SVGs were used in a row, each with 462×462 dimension, in total 640,332 pixels. Such pixel equality was maintained to obtain a fair comparison among the designs in the user studies.

6.2 JSON Data Input and Processing

For the visualizations, we created several synthetic and real-life JSON datasets using python data processing. Each JSON file contained 4 variables with a 200×200 dimension (200 width and 200 height). For synthetic data, no pre-processing was done before passing them to `d3.contours()` function. The real data went through few pre-processing steps as discussed in the data description section.

6.3 Contour Creation

We used `d3.contours()` function to get the contours. This function returns an array of objects. The size of the array is equal to the number of thresholds i.e. the number of contour intervals provided to the function. Each element of the object array has three keys: 'type', 'value', and 'coordinates'. For contours, the key 'type' returns 'MultiPolygon', 'value' returns the threshold associated with the contour and 'coordinates' contains the contours at that threshold. For example, if threshold array was $[0, 0.25, 0.5, 0.75]$, then 4 objects will be returned. The second object will have a 'value' of 0.25 and the coordinates will have the array/s of data points who altogether represent the isoline/s or contour path/s who separate the region having data points of value less than 0.25 from the region having data points of value from 0.25 to 0.5. There is also a concept of 'holes' in contours. Holes are regions of lower threshold value inside a higher threshold contour, like two circles with the same center and different diameter. If the outer circle has a higher threshold, the inner circle then is a hole. Thus, the coordinates of the hole is added as a nested entry to the higher threshold contour coordinates in stead of being added to its own array of coordinates. For example, let us assume that, for value 0.25, there are two contour paths. So, they would be added to the 'coordinates' array as `coordinates[0][0]` and `coordinates[1][0]`. Now, there is a hole inside the second contour path. As mentioned, the hole should have a lower threshold value than 0.25, which is 0 in this example. Therefore, the points array for the hole will be added to the coordinates of 0.25, as `coordinates[1][1]`. If there were two holes inside that contour, they would have been added as `coordinates[1][1]` and `coordinates[1][2]`.

6.3.1 Contour Simplification of Synthetic Data

The contour lines of the synthetic data were simplified before visualization. The scatter plot of A's clusters from Figure 6.2 (left) shows that the outer contour bands have a lot of loose data points around them, resulting in numerous cringes

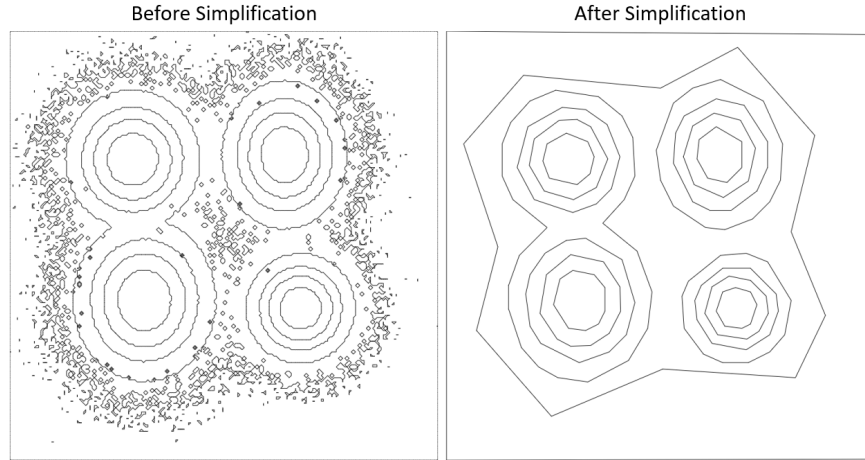


Figure 6.2: Synthetic data - contour simplification

after contour making. There are also some tiny contours at random positions generated by the function. So they had to be simplified using a simplification function. The function was fed two parameters- a polygon and a tolerance value; the higher the tolerance value, the aggressive the simplification becomes. The contours after simplification look like the B, C, or D contours of Figure 6.2 (right).

For real-world data, no simplification was done to keep the realness of the data. The contour lines found after this step was then passed into the visualization algorithms.

We now describe the details of the implementation. For ease of explanation, let b , c and d denote the normalized B, C, and D values, respectively.

6.4 Choosing A, B, C, and D

A good understanding of the variables' distribution is needed to decide A, B, C, and D, as a poor choice can lead to a sub-optimal visualization. The choice also depends on the task that the user will perform. A general rule of thumb would be, the variable that will be used as a base to interpret the variables' distribution should be chosen as A. For example, if the user wants to know how temperature, pressure, and humidity vary over soil moisture, then soil moisture can be encoded as A, i.e. the contour lines of soil moisture will be drawn to visualize the other three variables.

A sparse contour map (fewer number of contour lines) of A will not effectively visualize the other variables' distribution due to the lack of contour lines. On the other hand, a very dense contour map with lots of contour lines would clutter the map, distracting the user from interpreting data correctly. Therefore, a good choice of A is aided by a good choice of the number of contour thresholds as well as the threshold values (e.g. regular intervals, percentiles).

6.5 Visualization

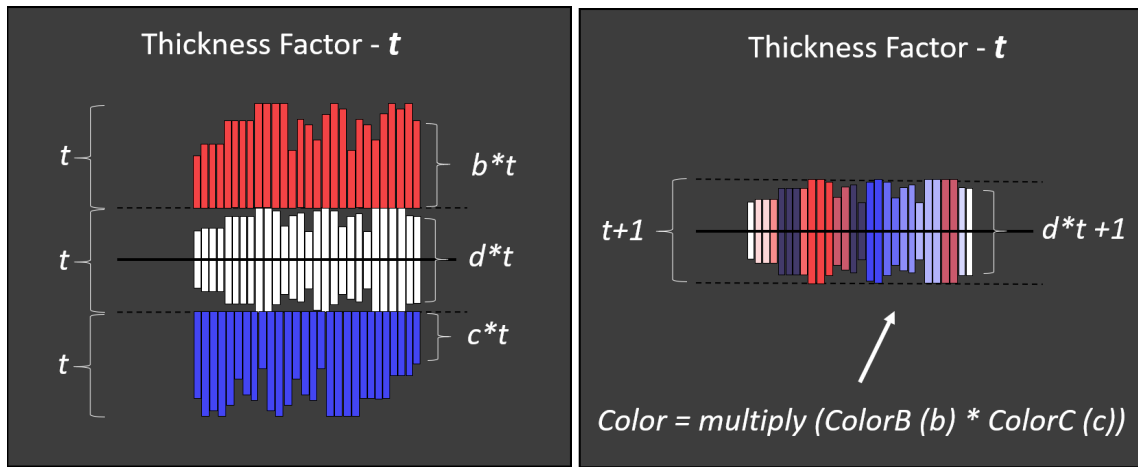
The contour lines were processed sequentially, from the lowest to the highest threshold. The steps of the visualization are given below:

1. **Border Control:** For a contour path, we had the array of data points. During this step, the contour lines that are very close to or on the display borders were just plotted normally. Usually, the contour function of any variable distribution returns few contours, i.e. the 4 borderlines of the display, and those lines were ignored. Also, for our 800×800 pixels SVG display, the lines that fall within the 5 pixels range from any border were ignored, i.e. 0-5 pixels from the top and left and 795-800 pixels from the bottom and right. These cases can be seen in the real-life visualizations (see the borders of Figure 3.4 visualizations).
2. **Polygon Area Filtering:** For real-life datasets, we did not visualize tiny contours. So, the contours enclosing an area of less than 10 unit squares were removed from the visualization.
3. **Line Segment Slicing:** The lines that fell inside the 790 pixels width and height range were then sliced into very tiny line segments. The amount of slicing is proportional to the length of the actual line. If a line is sliced once, 3 data points are found. This way, for x number of slices, $x+2$ data points on the line were considered. Then the values of B, C and D at those points were calculated and encoded into visualization accordingly. All the visualizations are done using *d3.path*.
4. **Calculating B, C, and D values:** To get the values of B, C and D at any point, we used a function called *getContourValue()* which returned the appropriate value of the contour interval containing the data point. For example, if the data point falls into B's contour interval of threshold value x and C's contour interval of threshold value y , then x and y are returned as B and C values respectively. The rest of the design-wise encoding schemes have been described below. For the purpose of explanation, let's assume that the B, C, and D values are denoted as b , c and d .
5. Visualizing B, C, and D on the contour lines of A

6.5.1 Encoding *Parallel Lines*

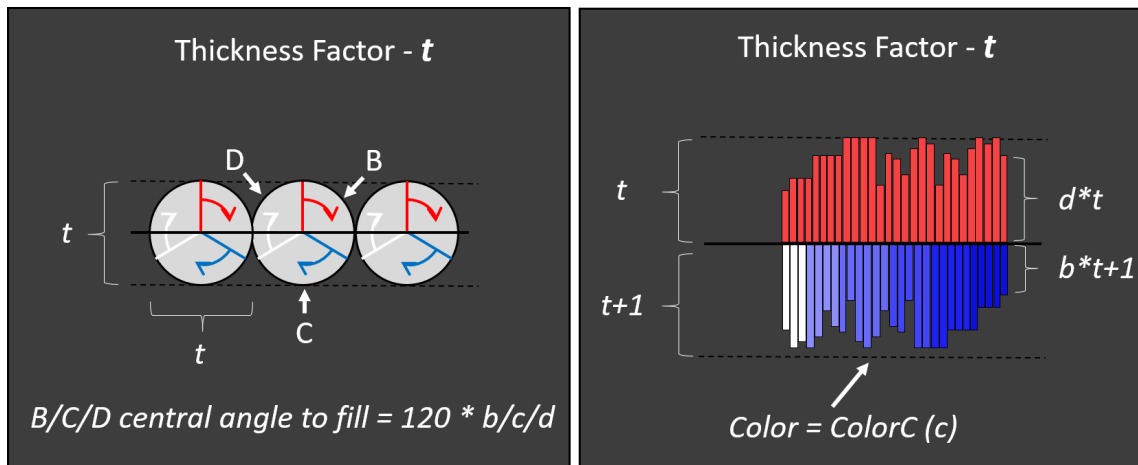
At every segment on A's contour line, B, C, and D were drawn with perpendicular lines. As illustrated in Figure 6.3(a), d were drawn on both sides of A's contour line, and B and C were drawn on opposite sides of D. A thickness factor t was used to linearly map the attribute values to the input line-thickness range. On a line, at every slicing point, perpendicular lines are drawn on both sides of A's contour lines where the perpendicular lines' heights are the multiplications of B or C values and the thickness factor t i.e. $b \cdot t$ or $c \cdot t$. D's lines have the same feature (line width $d \cdot t$) except that they intersect A's lines in the middle, half on B's side and half on C's.

If we look closely at Figure 6.3(a), we can see that B and C perpendicular lines start from the maximum possible width of D at any point, which is thickness factor t . This makes them all 3 close to each other and disjoint at the same



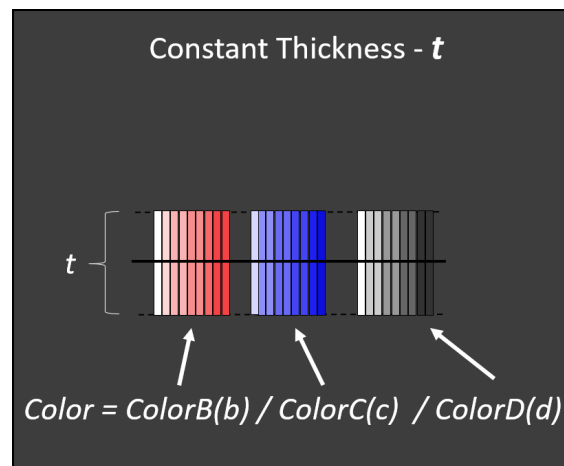
(a) Parallel Lines

(b) Color Blending



(c) Pie

(d) Thickness-Shade



(e) Side-by-Side

Figure 6.3: Illustration for the implementation details for different designs

time. Thus if all the attributes attain their maximum, then they appear to be adjacent to each other. The advantage of this is to get a clear perception of D's presence and absence anywhere along the contour line. At any point, the maximum possible width of all 3 variables combined is $3 \cdot t$ and the minimum is 0. When it is 0, one can only see the thin contour line of A, which has been done to visualize the continuity of A's contour lines.

6.5.2 Encoding *Color Blending*

Here the value of D determines the line width, where the colors of B and C are blended on the D's line. As shown in Figure 6.3(b), at every segment, one perpendicular line is drawn which intersects A's contour line in the middle. The perpendicular lines' height is $(dt + 1)$ pixels. Here the constant 1 ensures that the colors can be blended even when d is 0. For each of B and C, the colors were chosen using a discrete monochromatic color scheme with a perceptually uniform color distance. The number of discrete shades in the scale depends on the number of contour intervals. We blended the colors using the *mix-blend-mode* scheme *Multiply* of CSS, which was chosen in a pilot study with 63 participants on 3 possible candidate schemes: *Multiply*, *Darken* and *Difference*. The participants had to complete 24 value estimation tasks, 8 in each blend mode. Out of 3 candidates, *Multiply* had the highest mean accuracy (mean=67%, sd=47%) than *Darken* (mean=61%, sd=49%) and *Difference* (mean=64%, sd=48%).

6.5.3 Encoding *Pie*

In this design (Figure 6.3(c)), each attribute was allotted a 120° space to create a pie slice. A single pie has a number of sliced data points beneath it. Therefore, inside a pie, the value of any variable is the average of the values for all the sliced data points that the pie overlaps. After all 3 variables values are averaged, that value gets multiplied by 120° to get the central angle for filling up the area in a clockwise direction.

The center of the pie is located on the contour line of A. If $b = c = d = 0$, then a pie becomes an empty circle and the maximum values ($b = c = d = 1$) will cover the whole pie in 3 equal colored portions.

6.5.4 Encoding *Thickness-Shade*

This design visualizes B and D on opposite sides of the A's contour lines (Figure 6.3(d)). The D value is mapped to a perpendicular line of height $d \cdot t$. For B, the value is mapped to $b \cdot t + 1$, where a minimum of 1 pixel allows encoding C even if B is 0. The color for C is chosen using a discrete monochromatic color scheme with a perceptually uniform color distance. The number of discrete shades was based on the number of contour intervals.

The minimum width of all visualizations combined is 1 pixel and the maximum width is $b \cdot t + d \cdot t + 1$.

6.5.5 Encoding *Side-by-Side*

In this design, the perpendicular lines have the same fixed height t and intersect A's contour lines in the middle (Figure 6.3(e)). The B, C, and D values were encoded in the corresponding discrete sequential color scheme.

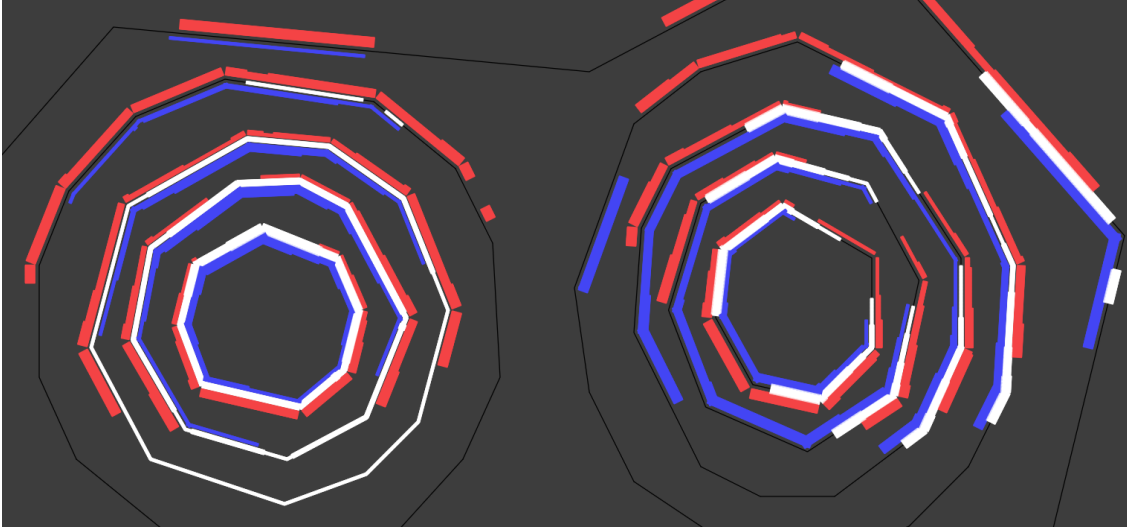


Figure 6.4: Line break (for B in red) in *Parallel Lines* (at line segment intersections) due to drawing perpendicular lines along a contour line

The colors from the sequential color scales of the variables were taken from a range of shades of the same color, chosen from a website [102].

In *Parallel Lines*, we can see that the red B lines have breaks (Figure 6.4) from one line segment to the next adjacent one. This happened because the perpendicular lines are drawn along the contour line points and at the intersection of two line segments, the red lines get drawn at an angle with respect to each other. This effect is not visible for C and D as they are drawn inside, unlike B. Such breakpoints might be found in *Color Blending* and *Thickness-Shade* as well. This might be addressed by filling the breakpoints up the tiny triangular shapes.

6.6 Evaluation

A careful choice for various design parameters is important to achieve optimal readability for the designs. Two major factors that can influence the designs are line width (the space allocated along the base contour line for encoding B, C, and D), and the number of contouring thresholds for A. Therefore, we first conducted two studies to examine these factors and choose appropriate parameter values for the designs. In the final study, we evaluated the designs based on the viewers' task performances. The first two studies were conducted on synthetic data and the final study included both synthetic and real-world data.

CHAPTER 7

STUDY 1 (CONTOUR WIDTH)

Our first study investigated the effect of contour width on each design’s interpretability, as well as on tasks that use the underlying map.

Intuitively, increasing contour width should make the encoded variables easier to see and interpret, but will also increase occlusion of the map; in addition, wide contours may also overlap each other, depending on the density and shape of the contours. To investigate this trade-off, we set the number of contour intervals to 8 (i.e., 7 contouring thresholds), and then determined a range of widths to explore for each design. We used 8 contour intervals because this allows designers a reasonable spectrum of design choices, and gives us a reasonable range, i.e., [0,8], to investigate in Study 2 (described below).

Table 7.1: Different widths for Study 1

Design	Width Alternatives		
	Minimum	Median	Maximum
<i>Parallel Lines</i>	9	-	12
<i>Color Blending</i>	4	7	10
<i>Pie</i>	8	-	10
<i>Thickness-Shade</i>	8	-	10
<i>Side-by-Side</i>	1	2	4

Table 7.1 illustrates the width ranges used in Study 1 (note that B, C, D were varied to cover all combinations). We chose minimum and maximum widths for each design based on informal testing with each design’s encoding. The minimum width is determined by the number of pixels required to create the design and make the variation in the attributes noticeable. For example, *Parallel Lines* requires 9 pixels to encode the variation (low, mid, high) for each of the three attributes. The maximum width corresponds to the case when the successive linear elements are about to overlap.

We also selected a median width if there was enough difference between minimum and maximum that the median would have at least two-pixel units from the extremes. Hence we only have the median width for *Color Blending* and *Side-by-Side*. For *Parallel Lines*, the widths are multiples of 3 as B, C and D have the same width. For *Pie*, the average

width in between the maximum (8) and minimum (10) does not have any significant difference in visualization. For *Thickness-Shade*, the widths are multiples of 2.

7.1 S1: Participants, Data, and Tasks

Table 7.2: Tasks with domains for Study 1

ID	Task	Domain
1	Select the marked contour region that best represents the following combination: high B, low C, and high D	Compare different marked contour regions
2	Consider the contour regions intersected by the lines. Select the directed line that best represents the following trend: B and D both increase, and C decreases	Interpret trends across contour lines
3	Select the marked contour region that, when moving clockwise, best represents the following trend: B decreases, D increases, and C stays the same	Interpret trends along a contour line
4	Count the number of lines that show the following: B and C have the opposite trend to D	Identify similar/ opposite trends across contour lines

We ran a crowdsourced study on Amazon Mechanical Turk (AMT) [30, 51, 62]. To be eligible, a participant needed to pass a color perception test (Ishihara test [22]), run the study on a desktop computer, reside in North America, and have at least an 80% approval rate in AMT. We recorded 126 responses, and among those 63 were complete responses (32 male, 31 female, median age range 30-39). Out of the final 44 participants, 22 self-reported familiarity with data visualization interfaces.

All experimental tasks were created using synthetic data. We tested the 5 designs described earlier, with the width choices determined for that design (12 in total, see Table 7.1). Participants completed 57 tasks (4 tasks for each width that involved interpreting the variables encoded in the design, plus one additional task for three of the widths that involved reading the background map).

7.1.1 Choosing the Tasks

The choice of tasks is very crucial to evaluate a design accurately through user studies. In this project, the very aim was to explore the effect of leveraging contour lines for visualizing three more variables on interpretability. So we listed the common geospatial data analysis task domains that cover a diverse set of geospatial data analysis tasks. One common task would be to identify the value of a variable (e.g. higher temperature) at a certain region, in our case, at a certain part of the contour line. In a multivariate case, this task can be extended to identifying a contour part with higher B, lower C,

and higher D where B, C, and D can be any three variables from the visualization. Another very common task is to identify the trend of a variables along a direction. This trend analysis task is widely done to examine the behaviour of a variable. In our case, trend can be found across multiple contour lines in a direction, or along a single contour line going clockwise or anticlockwise direction. For example, one would like to find a trend where B (e.g. pressure) is increasing, C and D (e.g. temperature and soil moisture) are decreasing along a direction (e.g. north-east to south-east).

Considering the feasibility of the study, it was not possible to add all the tasks into the study, which would lead to a very long session (almost 2 hours), increasing the participant's cognitive load and compromising the quality of data. Therefore, we summarized the tasks and chose four representative interpretations of tasks to be used for Study 1. In our four interpretation tasks (Table 7.2), one involved identifying values, two involved looking for trends, and one involved comparing trends (Figure 7.2). For tasks 1 and 3, we marked four contour sections on the design (Figure 7.1 left), and participants selected the option that matched the requested combination or trend. For tasks 2 and 4, we drew four lines across the contours, and participants selected the option that best identified a specific trend (e.g., Figure 7.1 right).

The background map reading task used the *Color Blending* design with 3 of the widths (4, 8, and 12). In this task, icons were placed in the underlying map, and participants were asked to count the number of icons and select an answer from 4 options.

The reason for choosing *Color Blending* as a representative design for the background task is that it has three different width choices that are substantially different from each other. The icons were 8×8 pixels. The number of icons ranged between 18 to 22, and were placed randomly on the map; this meant that in some cases the icons were partially hidden by the contour lines.

All the tasks were multiple choice questions where a participant had to select one of four choices. Each question had only one answer that the participant had to find out from the visualization. Figure 7.1 shows that each interpretation task image had four markers on them, 1 to 4. Task 1 and 3 had markers marking four contour parts, task 2 had four arrows and task 4 had four lines drawn on the contour lines. Task 1, 2 and 3 were straightforward, find the answer and choose the marker number as the answer. Task 4 was to count a trend, which would be from one to four. For the 9 background tasks, participants had to count the map icons and select the accurate number from four choices. We varied the four choices from a pool of five (among 18, 19, 20, 21, and 22).

7.2 S1: Hypotheses

We had 3 hypotheses for this study, as mentioned in Table 7.3.

7.3 S1: Procedure

The procedure of Study 1 is given in Figure 7.2. As the study was online in nature, we restricted the devices to do the study to laptops and desktops (devices with larger displays). Then the participants completed an informed consent form and completed a color blindness test, as our designs had different colors in encoding. The participants who passed the color blindness test were then shown a description of the five novel visualization alternatives starting with the basics of

Table 7.3: Study 1 hypotheses

ID	Hypothesis
h_1	If width increases, accuracy will increase and completion time will decrease
h_2	The influence of width will be more noticeable for <i>Parallel Lines</i> , <i>Pie</i> , and <i>Side-by-Side</i> than for other designs (because <i>Color Blending</i> and <i>Thickness-Shade</i> could be more difficult to interpret for inexperienced users)
h_3	For the background task, increasing contour width will lead to lower accuracy and higher completion time

contour maps. After the design demonstration, they were given a set of practice tasks to complete. After each practice task, a participant was told whether the response was correct or not, and were given an explanation of the correct answer with a brief justification.

Finally, after doing the practice tasks, the main study tasks were given. The interpretation tasks came first and then the background tasks. Interpretation asks were grouped according to design and thus, there were five groups for five designs. After going through all the tasks of each design, participants completed a NASA Task Load Index (NASA-TLX) style effort questionnaire [44]. After all the five groups of the 48 interpretation tasks, they did 9 background tasks. After completing the background tasks, i.e. at the end of the study, they rated their familiarity with visualization interfaces and their preferences for each of the widths.

Each participant completed the 57 tasks as described above ([12 width choices X 4 tasks = 48] + [3 backgrounds X 3 intervals = 9]). Participants were asked to complete the tasks as quickly and accurately as possible. Each task started with a ‘start’ button, and ended when the participant selected an answer option and pressed ‘next’. Before starting a new task, participants were shown a reminder that they could rest before continuing.

The study used a within-participants design, with contour width as the independent variable (considered separately for each design); dependent variables were accuracy, completion time, and subjective effort scores. The independent variable, i.e., factor was width and the dependent variables were accuracy, completion time, and subjective responses.

Designs and tasks were presented in random order (sampling without replacement). Due to the online nature of the study, two or more participants doing the study concurrently completed the designs in different orders, but, the number of possible responses were unknown and it was not possible to check the completeness of current responses and change the order accordingly, on the fly, to maintain a Latin square order.

7.4 S1: Results

We applied additional filters to test whether participants were legitimately attempting the tasks: first, we removed participants who did not have an overall accuracy of 30% (this is slightly higher than chance based on our four-question

tasks); second, we removed participants who had large time gaps in their tasks or surveys. The rationale behind accuracy filtering this is, the probability of randomly selecting an answer and getting it right is 25% for a 4 options multiple choice study. Therefore, to ensure better data quality, the accuracy filtering was done. After filtering, we had 44 participants (20 male, 24 female) with a median age range of 31-39. The average time for completing 57 tasks was 29 minutes, 65 minutes being the maximum.

7.4.1 S1: Interpretation tasks

For the four interpretation tasks involving the B, C, and D datasets, data were analyzed using repeated-measures ANOVAs for each design (because each design used a different set of widths); Bonferroni corrected paired t-tests were used for follow-up comparisons. Since the designs had different number of width choices, We examined the effect of width type on completion time and accuracy for each design. Figure 7.3 (a) shows that accuracies increased slightly as contour width increased, and Figure 7.3 (b) shows that completion times decreased overall as width increased.

We found significant effects of width on accuracy for the *Parallel Lines* design, and on completion time for *Color Blending*, *Pie*, and *Side-by-Side*. No effect of width was found for *Thickness-Shade* in accuracy ($F_{1, 43} = 0.025$, $p = .23$) or completion time ($F_{1, 43} = 1.44$, $p = .87$). For *Parallel Lines* (using widths 9 and 12), we found a significant effect of width on accuracy ($F_{1, 43} = 5.4$, $p < .05$), with width 12 having higher accuracy than width 9. For *Color Blending* (widths 4, 7, 10), we found a significant effect of width on completion time ($F_{2, 86} = 8.09$, $p < .05$); Figure 7.3 (b). Post-hoc t-tests showed that width 10 was faster than both width 7 and width 4 (all $p < .05$). For *Pie* (widths 8 and 10), we found a significant effect of width on completion time ($F_{1, 43} = 9.46$, $p < .05$). Width 10 was faster than width 8. For *Side-by-Side* (widths 1, 2, 4), we found a significant effect of width on completion time ($F_{2, 86} = 13.46$, $p < .05$). Post-hoc t-tests showed that widths 2 and 4 were faster than width 1 (all $p < .05$).

7.4.2 S1: Background Task

The background icon-counting task used design *Color Blending* with widths 4, 8, and 12. We found a significant effect of width on accuracy ($F_{2, 86} = 121.76$, $p < .05$). Post-hoc t-tests showed significant differences among all 3 widths ($p < .05$). As shown in Figure 7.4, the mean accuracy for width 4 was higher than 8, which was higher than that of 12. There was no effect of width on completion time ($F_{2, 86} = 0.85$, $p = .43$).

We also asked users for their preferences of widths in the survey and found out that more users preferred the higher widths (Figure 7.5).

7.4.3 S1: Subjective Effort and Preference

We asked participants to rate their amount of mental effort, overall effort, frustration, and perceived success with each design. Friedman tests showed significant differences in all questions (all $p < .005$), with *Parallel Lines* and *Side-by-Side* rated better than the other designs. The mean scores and the Friedman test are shown on Figure 7.6 and Table 7.4. The width preference question reveals higher user preferences for the maximum (50% of participants) and median (41%)

Table 7.4: Study 1: NASA-TLX scores

	χ^2	p
Mental Effort	35.4	3.9 e-7
Perceived Success	87.8	2.2 e-16
Hard Work Required	16.6	.002
Annoyance / Frustration	20.8	.0003

Table 7.5: Width choice after Study 1

Design	Width
Parallel Lines	12
Color Blending	10
Pie	10
Thickness-Shade	10
Side-by-Side	4

widths.

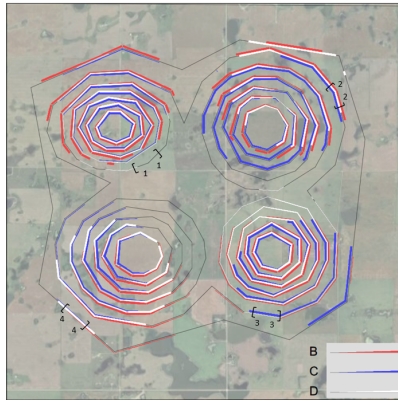
7.5 S1: Discussion

Increased contour widths for interpretation tasks led to improved completion time in three of the designs, and improved accuracy for one design, partially supporting hypothesis h_1 . We did not find any significant effect of width for *Thickness-Shade*, which partially supports our hypothesis h_2 that suggested the effects of width would be more obvious for some designs. Our results for the background task (effect of width on accuracy, but not on completion time) partially support hypothesis h_3 . Overall, the fact that there was only a minor effect of reduced width on interpretability (particularly for accuracy) means that width can often be safely reduced in scenarios where the visibility of the background is critical.

7.6 S1: Summary

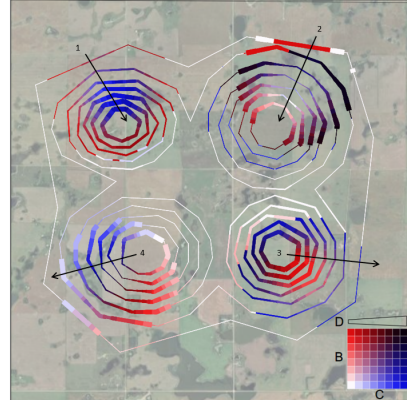
We chose 1 width alternative for each design based on the data analysis, which is shown in Table 7.5. This width has been used later in Study 2 for all 3 contour interval variations- 4, 6 and 8. In the following section, we explore the influence of contour intervals, which is another important element of a contour plot.

Please select the contour part that best represents the combination:
lower B, lower C, and higher D



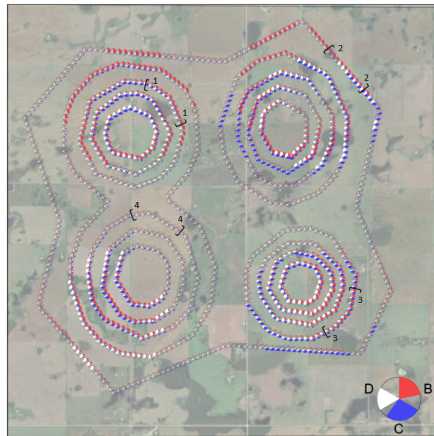
(a) Task 1: Value identification

Please consider the contour parts intersected by the given lines. Select the
directed line that best represents the trend: B and D **decrease**, and C **increases**



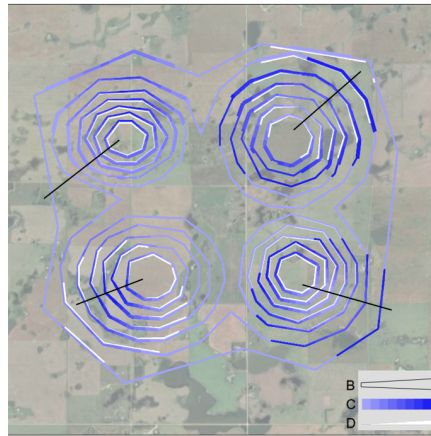
(b) Task 2: Looking for a trend (across contour lines)

Please select the contour part, considering clockwise direction, that best
represents the trend: C and D **increase**, and D **stays the same**



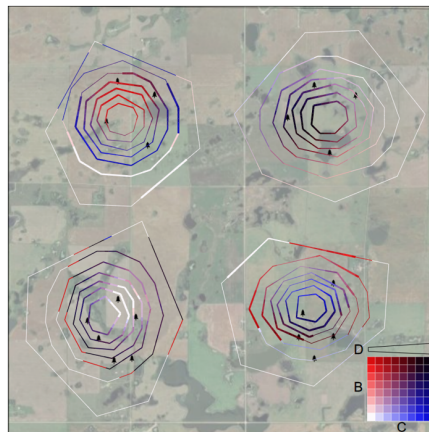
(c) Task 3: Looking for a trend (along a contour line)

Please count the number of marked lines that represent
the following: B and C have the **similar** trend to D



(d) Task 4: Comparing among trends and counting one

How many icons are there on the map
[icons are small and black in color]?



(e) Task 5: Counting background landmarks

Figure 7.1: Illustration for interpretation (task 1-4) and background (task 5) tasks

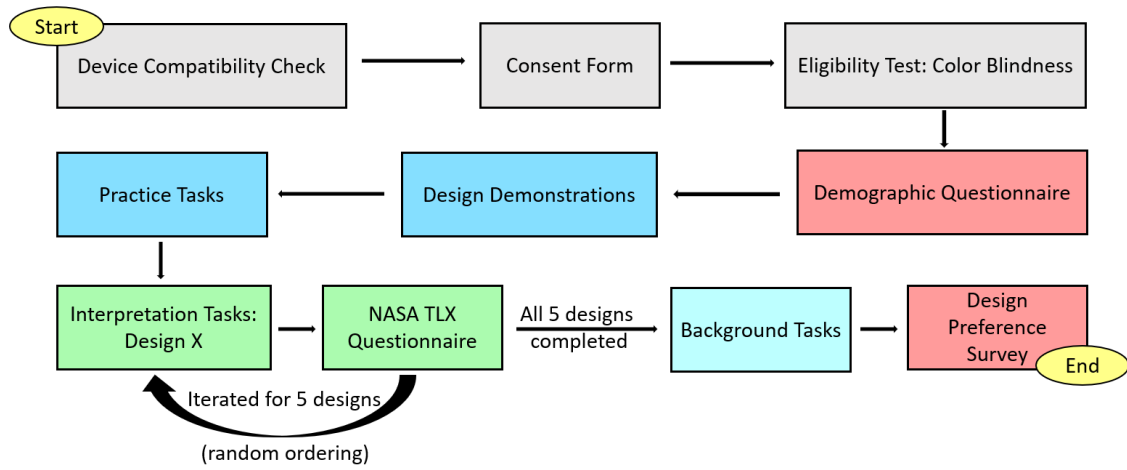
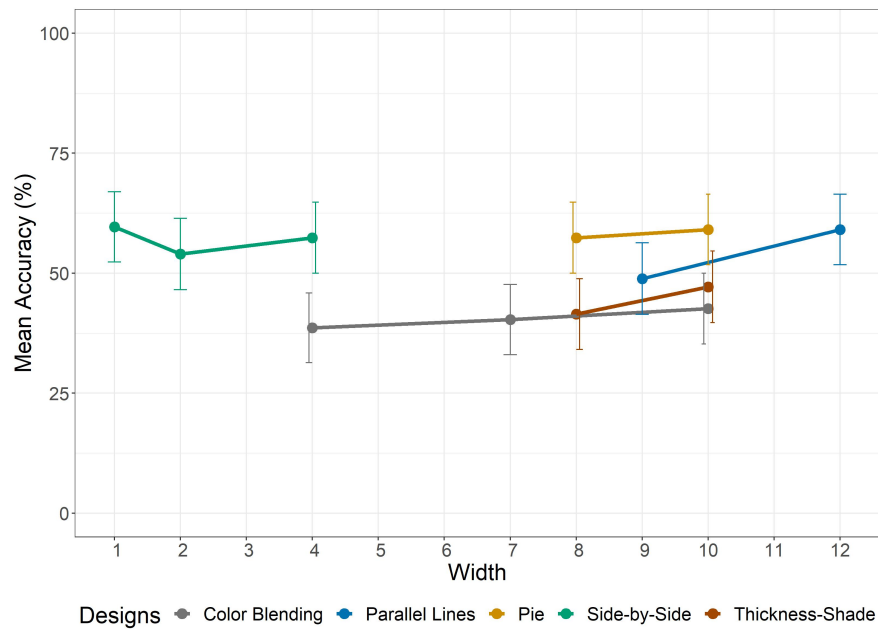
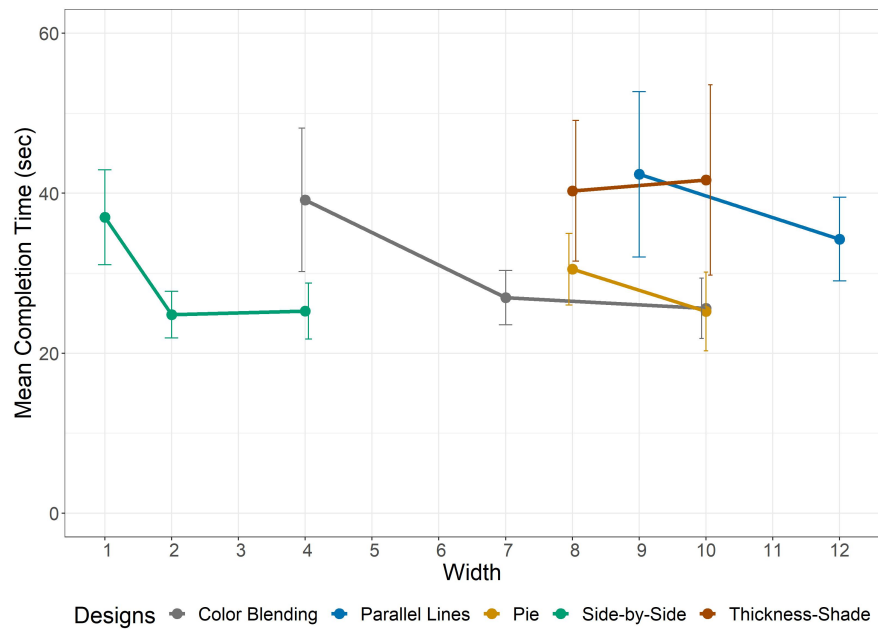


Figure 7.2: Study Procedure



(a) Width VS accuracy (tasks 1-4)



(b) Width VS completion time (tasks 1-4)

Figure 7.3: Study 1: width VS accuracy and completion time

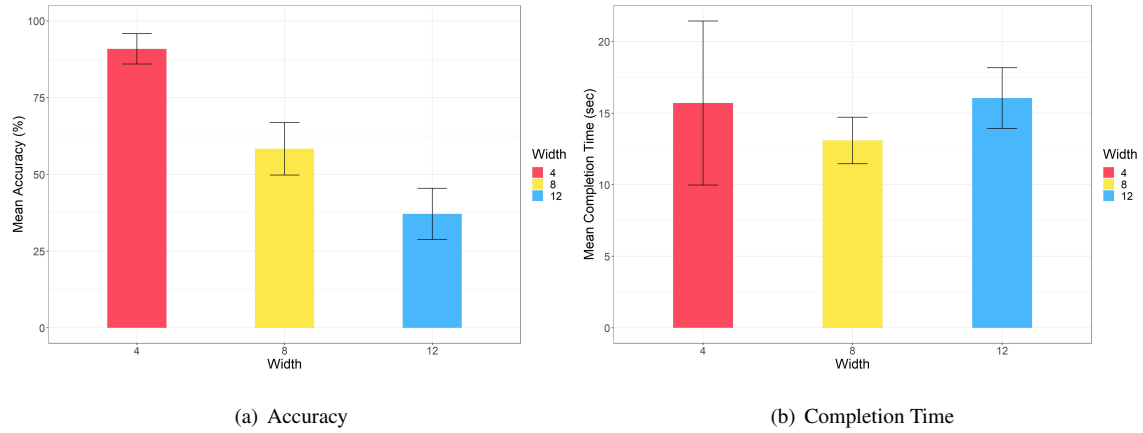


Figure 7.4: Study 1: background task

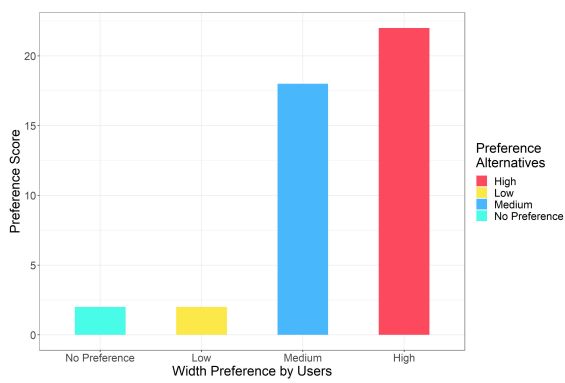


Figure 7.5: Study 1: width preference survey

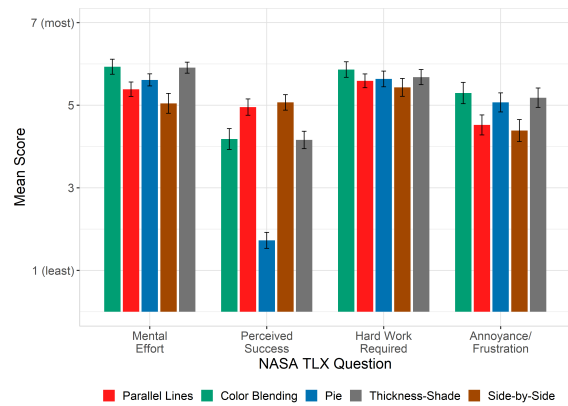


Figure 7.6: Study 1: NASA-TLX scores

CHAPTER 8

STUDY 2 (CONTOUR INTERVALS)

A higher number of contour intervals increases both the number of visual elements in the design and the degree of background occlusion. Increasing the number of contours, however, also provides more data points for the other variables visualized on the contour, and so may increase the interpretability of these variables. Our study explores this trade-off using a study design similar to that used above.

8.1 S2: Participants, Data, and Tasks

We ran the study on Amazon Mechanical Turk with the same eligibility criteria as in Study 1. The procedure was kept the same too. We recorded 68 complete responses (40 male, 26 female, 1 non-binary, 1 preferred not to answer), aged 21-60 (median age range 21-29). None of the participants took part in Study 1. Out of final 46 participants, 36 self-reported familiarity with data visualization interfaces.

The study used the 5 designs described above, each with 3 contour interval alternatives (4, 6, or 8 contours). Participants completed 69 tasks: the same 4 interpretation tasks from Study 1 for each combination of design and contour interval, as well as the background icon-counting task ($[5 \text{ designs} \times 4 \text{ tasks} \times 3 \text{ intervals} = 60] + [3 \text{ backgrounds} \times 3 \text{ intervals} = 9]$).

For analysing the interpretation tasks, the study used a within-participants design with three factors: Design (the five designs described above), Task (the four interpretation tasks from Study 1), and Number of Intervals (4, 6, or 8). The main dependent measures were accuracy and completion time; we also collected subjective effort and preference scores.

8.2 S2: Hypotheses

We had 2 hypotheses for this study, as mentioned in Table 8.1.

8.3 S2: Procedure

Similar to Study 1, participants went through the eligibility tests, design demonstration, and practice tasks. Then they completed the main study tasks and filled out the TLX-style effort surveys after each design and overall preference questions at the end of the study. The data and task domains were the same as in Study 1, and designs and tasks were presented in random order.

Table 8.1: Study 2 hypotheses

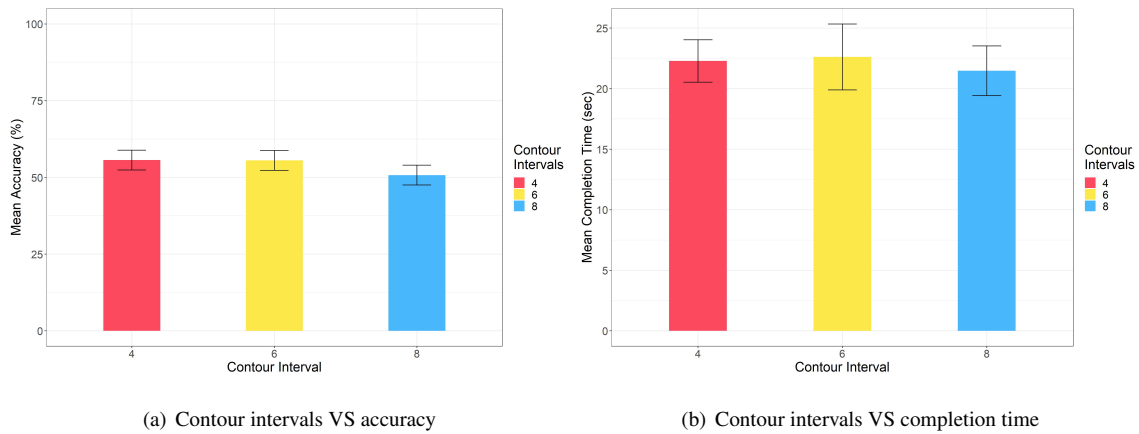
ID	Hypothesis
h_4	More contour intervals will result in better performance for the tasks that require analysis across contour lines
h_5	For the background task, more contour levels will lead to lower accuracy and higher completion time due to increased occlusion

Participants were briefed about the visualization alternatives and the basics of contour maps, followed by a set of practice tasks like Study 1. The main study had 69 tasks, five TLX-style surveys, one for each design, and a final survey. Each task was a multiple choice question and participants were allowed to take breaks in between every 2 questions.

The study used a repeated measures within participants design. The independent variables we selected were design, contour interval and task. The dependent variables were accuracy, completion time, and subjective responses.

8.4 S2: Results

After filtering the participants based on response consistency and accuracy threshold of $\geq 30\%$, we had 46 participants (28 male, 1 non-binary, 17 female) with median age range of 31-39. The average time for completing 69 tasks was 24 minutes, 53 minutes being the maximum.

**Figure 8.1:** Study 2: contour intervals VS accuracy and completion time

8.4.1 S2: Interpretation tasks

We carried out $5 \times 4 \times 3$ RM-ANOVAs (Design \times Task \times Number of Intervals) for both accuracy and completion time, with Bonferroni-corrected t-tests as follow-up. There was a significant main effect of Intervals ($F_{2, 90} = 3.69$, $p < .05$) on

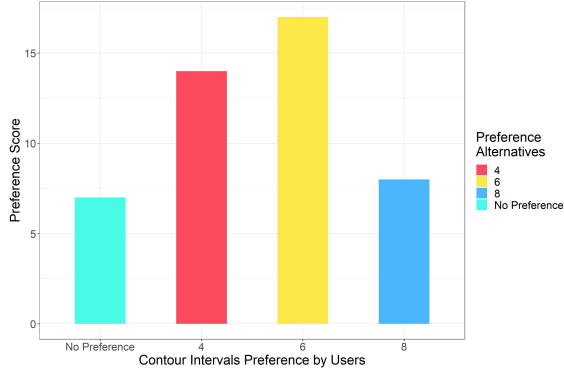


Figure 8.2: Study 2: contour intervals preference survey

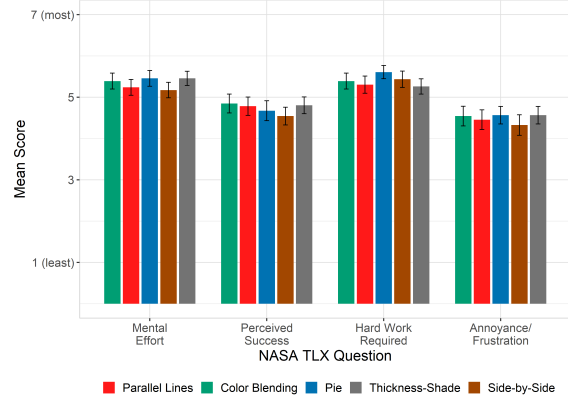


Figure 8.3: Study 2: NASA-TLX scores

accuracy. Post-hoc tests showed that 4 and 6 contour intervals had higher accuracy than 8 intervals (see Figure 8.1). There was also an interaction between Design and Task ($F_{12, 540} = 2.09, p < .05$). As can be seen in Figure 8.4(a), the *Pie* design had higher accuracy in Tasks 2 and 3 compared to the other designs. There was no main effect of Design on accuracy ($F_{4, 180} = 1.58, p = .18$). For completion time, there were no main effects of Design ($F_{2, 90} = 1.1, p = .36$) or Intervals ($F_{2, 90} = 0.33, p = .72$) (Figure 8.4(b)), but there was an interaction between Intervals and Task ($F_{6, 270} = 3.21, p < .05$).

8.4.2 S2: Background Task

We found no main effects of the number of contour intervals on either completion time ($F_{2, 86} = 0.65, p = .52$) or accuracy ($F_{2, 86} = 3.05, p = .05$) for the icon-counting task.

Table 8.2: Study 2: NASA-TLX scores

	X^2	p
Mental Effort	8.36	.08
Perceived Success	.8	.94
Hard Work Required	16.58	.002
Annoyance / Frustration	1.12	.891

8.4.3 S2: Subjective Effort and Preferences

Participants rated mental effort, overall effort, frustration, and perceived success with each design. Responses were similar across all designs, and Friedman tests showed a significant difference only for overall effort ($p < .05$), with the

Pie design seen as requiring more effort than the others. The mean scores and the Friedman test are shown on Figure 8.3 and Table 8.2. We also asked participants about the number of intervals they preferred, which reveals that users preferred 4 (36%) and 6 (39%) intervals to 8 (Figure 8.2).

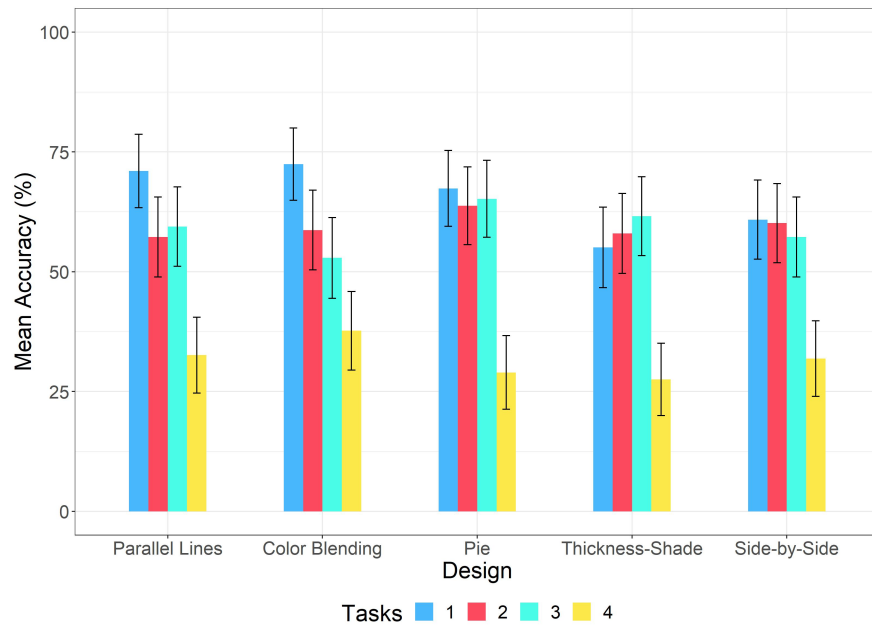
8.5 S2: Discussion

Overall, 4 and 6 intervals performed better than 8. One main reason for this result is that fewer contour intervals result in less visual clutter. Although there were significant effect of contour levels on accuracy only for Task 3, there was significant effect between contour intervals and task for completion time, which partially supports h_4 .

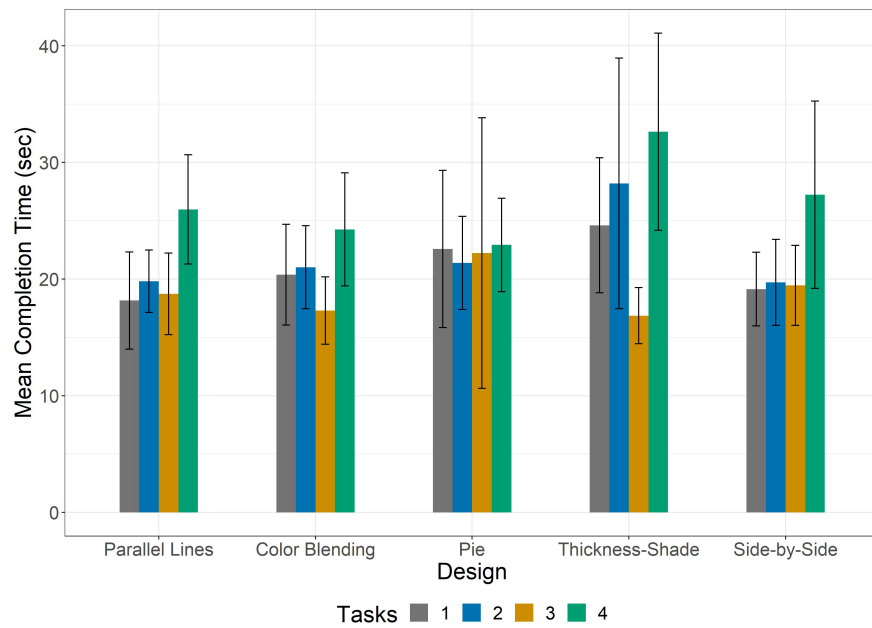
We observed that higher contour levels slightly reduced mean accuracy for Tasks 1 and 3, where users may have found it difficult to follow along a line where there were other parallel lines in close proximity.

The interactions show that task performance is dependent on design, contour intervals, and task combinations. The significant interaction (for completion time) between contour levels and tasks suggests that the association between contour level and designs depends on tasks. Similarly, for accuracy, the association between design and contour level depends on tasks.

In the next study, we focus on how performance varies by design in both synthetic and real world datasets. We used 4 contour intervals for all the designs.



(a) Accuracy



(b) Completion Time

Figure 8.4: Study 2: task accuracy for the five designs

CHAPTER 9

STUDY 3 (DESIGN COMPARISON)

After choosing the contour intervals and line widths for each design from the previous studies, the final study aimed to create a task specific design recommendation table to help users get the appropriate data visualization for a specific task domain.

For this study, we used both synthetic and real-life data. Unlike like synthetic, the real-life data pattern can vary arbitrarily over a map, which has the potential to reveal more insights to the way users detect and interpret such data.

9.1 S3: Participants, Data, and Tasks

We again ran the study on Amazon Mechanical Turk with the same eligibility criteria as in Study 1. We recorded 78 complete responses (41 male, 33 female, 2 non-binary, 1 preferred not to answer), median age range 30-39. None of these participants took part in Study 1 or 2.

We used two datasets for this study (one synthetic and one from a real-life scenario). Participants completed 6 different tasks for each design and dataset combination, which resulted in 60 tasks ([5 designs X 6 tasks X 2 datasets = 60]). Of the 6 tasks, 4 were similar to studies 1 and 2. We added 2 additional tasks (Table 9.1 and Figure 9.1) to examine whether users are able to interpret the extent of value changes of a single attribute (Task 5), and to estimate the difference between two attributes (Task 6). As all the first 4 tasks dealt with 3 variables, we also wanted to validate the designs' performance for tasks consisting of 1 or 2 variables.

Similar to Study 1, participants went through the eligibility tests, design demonstrations, and practice tasks. Then they completed the main study, the effort survey, and preference questions (in this study, participants stated their preference for one of the designs for each task, as well as overall).

9.2 S3: Procedure

The study used a within-participants design with three factors: Design (the five designs described above), Task (the six interpretation tasks), and Dataset (Synthetic or real-life). The main dependent variables were accuracy, completion time, and we also collected subjective effort and preference scores. Designs and tasks were presented in random order (sampling without replacement). The study flow was similar to Study 1 and 2 except that there was no background task and the participants were asked to rate designs in two ways- overall and task wise, at the end of the study.

Table 9.1: Tasks with domains for Study 3

ID	Task	Domain
1	Please select the contour part that best represents the following combination: Lower B, Lower C and Higher D	Compare different contour parts
2	Please consider the contour parts intersected by the given lines. Select the directed line that best represents the following trend: B increasing, C and D decreasing	Search for a trend across contour lines
3	Please select the contour part, considering clockwise direction, that best represents the following trend: B and C decrease, and D increases	Search for a trend along a contour line
4	Please count the number of marked lines that shows the following trend: B and C have a similar trend to D	Identify trends among variables, Count
5	Select the marked contour region that has the maximum change in (a given attribute)	Estimate value changes along a contour line
6	Select the marked contour region that has the minimum difference between (a given pair of attributes)	Estimate value difference along a contour line

9.3 S3: Results

After filtering based on response consistency and an accuracy threshold of $\geq 30\%$, we had 54 participants (30 male, 22 female, 2 preferred not to answer) with a median age range of 31-39; 45 participants reported familiarity with data visualization interfaces. The average time for completing 60 tasks was 30 minutes, 62 minutes being the maximum.

We carried out $5 \times 6 \times 2$ RM-ANOVAs (Design \times Task \times Dataset) for both accuracy and completion time, with Bonferroni-corrected t-tests as follow-up. There was a significant main effect of Design ($F_{4, 212} = 19.2, p < .001$) on accuracy.

Post-hoc t-tests showed that *Parallel Lines*, *Pie*, and *Side-by-Side* were significantly more accurate than *Color Blending* and *Thickness-Shade* (Figure 9.2). There was also a main effect of Dataset ($F_{1, 53} = 13.8, p < .001$): participants were more accurate with the real-life dataset (66%) than with the synthetic dataset (59%).

There were also significant interactions between Design and other factors. First, there was a Design \times Dataset interaction ($F_{4, 212} = 10.7, p < .001$). As shown in Figure 9.3, the *Color Blending* design had substantially lower accuracy for the synthetic data compared to the other designs, and only *Parallel Lines* was equally accurate with both datasets. Second, there was a Design \times Task interaction ($F_{20, 1060} = 5.01, p < .001$). Figure 9.3 shows substantial differences in the tasks depending on the design: for example, the accuracy of the *Color Blending* design was substantially lower in Tasks 1 and 6, and the accuracy of *Thickness-Shade* was lower for Task 6.

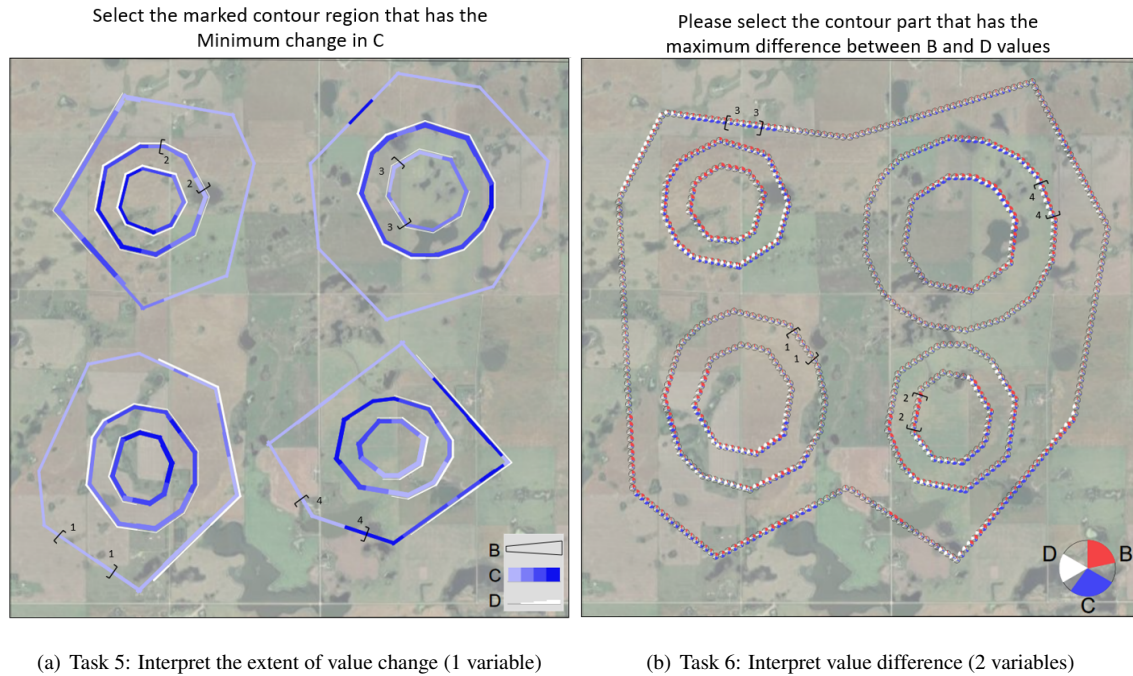


Figure 9.1: Study 3: Two additional tasks

For completion time, there were no main effects of Design ($F_{4, 212} = 1.39, p = .24$), but there were again interactions between Design and Dataset ($F_{4, 212} = 2.91, p < .005$) and between Design and Task ($F_{20, 1060} = 2.17, p < .001$).

9.3.1 S3: Subjective Effort and Preferences

We again asked participants to rate mental effort, overall effort, frustration, and perceived success with each design. Friedman tests showed significant differences for all questions ($p < .05$), with *Parallel Lines* and *Side-by-Side* scoring better than the other designs.

They also answered NASA-TLX score questionnaire after completing all tasks of each design. The mean scores and the Friedman test are shown on Figure 9.5 and Table 9.3 where Friedman test on each of the measures revealed significance.

We also asked users to rate the designs on a 0-4 scale (0: ‘not preferred’ to 4: ‘highly preferred’). Participants rated the designs for each task as well as their overall preference. Mean participant scores are presented in Table 9.2 and Figure 9.4 (higher values are better). The table highlights the top scores for each task and the top two designs for overall preference (*Parallel Lines* and *Side-by-Side* were the most-preferred designs).

9.4 Overall Discussion

The main finding of the third study is that all of the designs were successful for at least some of the tasks, and that the designs were similar in their performance – with the exception of *Color Blending*, which showed reduced accuracy

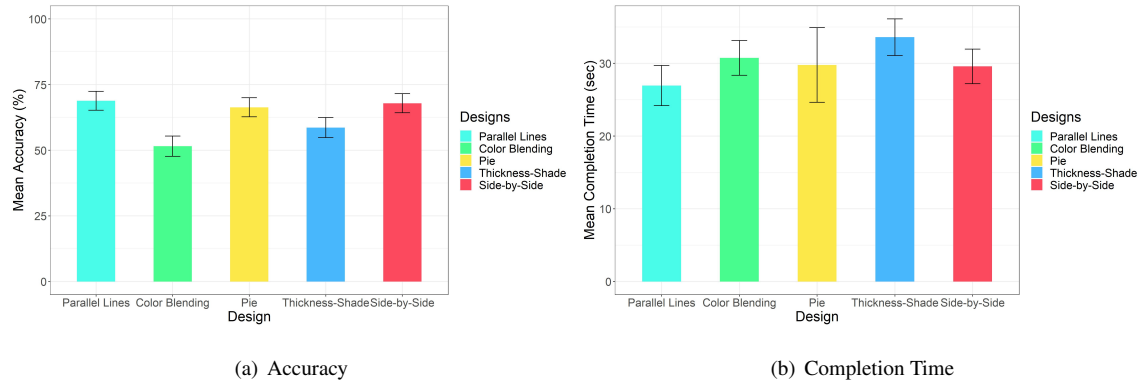


Figure 9.2: Study 3: overall performance of the different designs

Table 9.2: Study 3: design preference survey

Design	Average Preference Scores						Overall
	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	
<i>Parallel Lines</i>	2.28	2.5	2.48	2.37	2.57	2.74	2.85
<i>Color Blending</i>	1.85	1.83	1.87	1.57	1.76	1.93	1.93
<i>Pie</i>	1.78	1.87	2.24	1.72	2.06	2.02	2.15
<i>Thickness-Shade</i>	2.04	2.09	2.09	1.87	2.23	2.03	2.17
<i>Side-by-Side</i>	2.69	2.74	2.85	2.37	2.69	2.59	2.93

compared to the other designs for Tasks 1 and 6. The study also clearly showed that integrating multiple variables into a single contour line results in visualizations that users can interpret successfully – as successfully as separate individual presentations (*Side-by-Side*). This is an important result for situations where designers need to provide a single larger view rather than divide the available space into pieces, as is needed for a side-by-side presentation.

Overall, two designs—*Parallel Lines* and *Side-by-Side*—performed best and had high preference scores. Both designs have separate encoding space for all the variables; in addition, the encoding for different attributes was similar, and they are intuitive to read without any close inspection of the legend. *Side-by-Side* had high preference scores for all tasks except Task 6: in this task, a higher mean preference for *Parallel Lines* is likely due to its symmetric encoding for all the attributes, which makes the value difference easier to estimate, whereas for *Side-by-Side* users need to compute the difference inspecting two separate views.

Interestingly, however, both the *Parallel Lines* and *Side-by-Side* designs have space constraints (*Parallel Lines* in terms of contour width, and *Side-by-Side* in terms of display space). For scenarios where a single view is required and the visibility of the background is important, neither of these designs may be feasible. In these cases, the *Pie* design

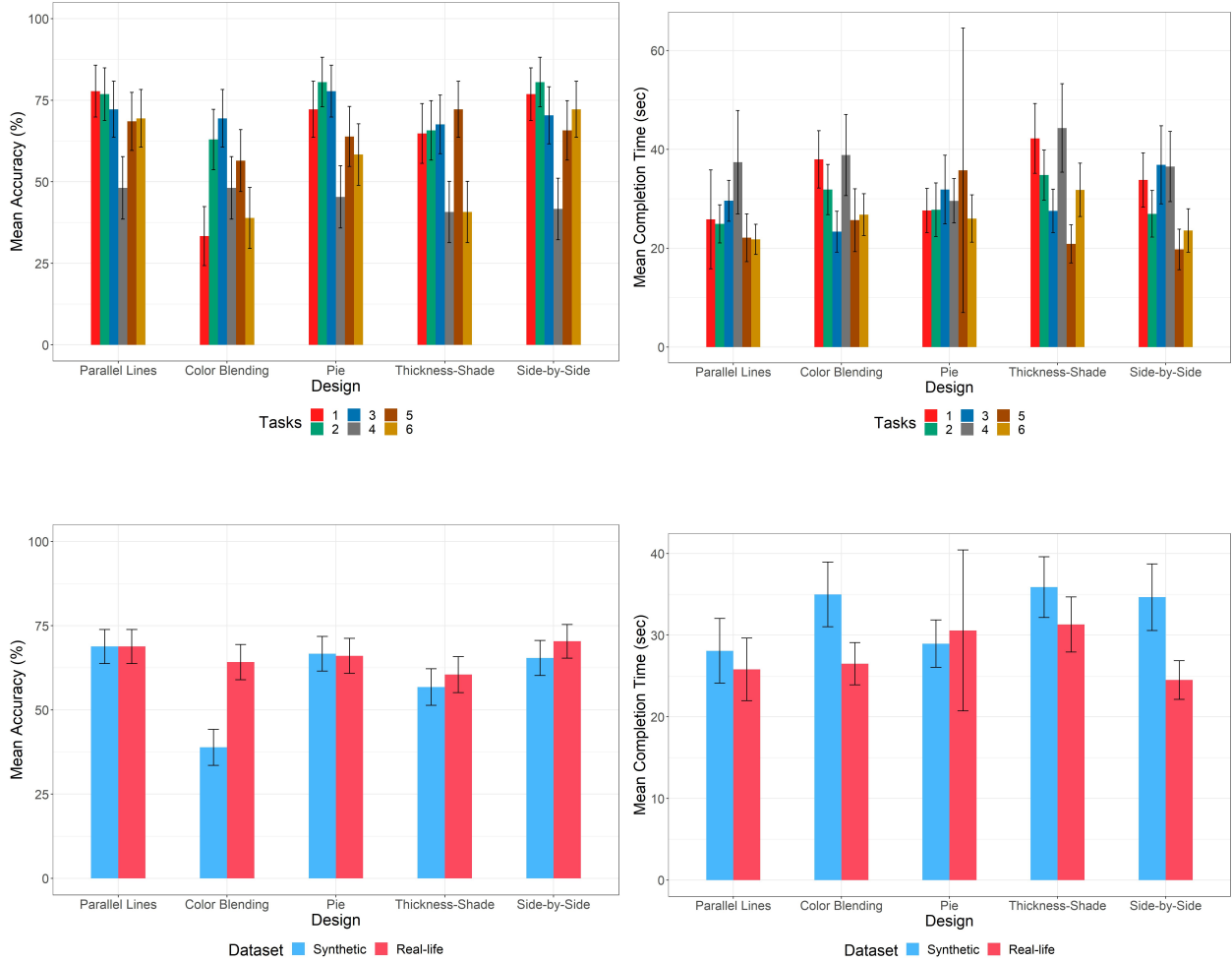


Figure 9.3: Task performances in Study 3, for (top) different designs, and (bottom) different datasets

appears to be a reasonable compromise because it had good accuracy and takes less space.

The *Color Blending* and *Thickness-Shade* designs both had poor performance on at least one task. This could be due to participants' unfamiliarity with the encoding, but the visual variables used in these designs may be more difficult to interpret overall. In addition, the encoding in these two designs was not symmetric compared to the other designs. Problems in Task 1 (for *Color Blending*) may have resulted from the need to estimate attribute value combinations, where the interpretation of the designs likely demanded a close inspection of the legend. In addition to the possibility of misinterpretation, this may have led to increased cognitive load or reduced effort for tasks using these designs. The same reasoning holds for the poor performance of *Thickness-Shade* for Task 6, where estimating the value difference between a pair of attributes encoded in different features such as thickness and color shade requires a careful reading of the legend.

Our results showed better performance with the real-life data than with the synthetic dataset (Figure 9.3), which may be due to differences in the underlying data distributions. The synthetic dataset consisted of almost all possible trend combinations for various attributes, so the corresponding visualizations consisted of highly varied feature combinations;

Table 9.3: Study 3: NASA-TLX scores

	X^2	p
Mental Effort	16.79	.002
Perceived Success	21.82	.0002
Hard Work Required	16.58	.002
Annoyance / Frustration	18.12	.001

in contrast, the visualizations generated from real-life data had fewer variations. It is likely that visualizations with real-life datasets are visually simpler, leading to improved accuracy and task completion time. These differences partially explain the significant interactions among design, dataset, and task combinations in accuracy, and interaction between design and dataset in completion time.

9.4.1 Task-wise Analysis

Table 9.4: Task-wise analysis table: Completion Time

Design	Completion Time (seconds)					
	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6
<i>Parallel Lines</i>	25.8	25	29.6	37.4	22.1	21.8
<i>Color Blending</i>	38	31.9	23.4	38.8	25.7	26.8
<i>Pie</i>	27.6	27.8	31.9	29.6	35.8	26
<i>Thickness-Shade</i>	42.2	34.8	27.6	44.3	20.9	31.9
<i>Side-by-Side</i>	33.8	27	42.4	36.5	19.8	23.6

Table 9.4 and 9.5 shows task-wise breakdown of the mean completion time and accuracy of all 6 tasks.

Task 1 involves comparing the values of all 3 variables. Among the 5 design alternatives, *Parallel Lines*, *Pie* and *Side-by-Side* have same encoding for B, C and D. An assumption is- designs with the same encoding for all 3 variables should have lower completion time with higher accuracy. The data analysis also reveals something similar to the assumption- *Parallel Lines* is faster than all designs except *Pie* and *Pie* is faster than *Thickness-Shade*. In accuracy, the reason behind all 4 designs' performing better than *Color Blending* can be that it is tough to identify the values of B and C from color when they are almost the same (along the diagonal of the design's B and C legend). Again, when D is low, the contour line width becomes the minimum, making it difficult to see the color and identify the B and C values.

Table 9.5: Task-wise analysis table: Accuracy

Design	Accuracy (%)					
	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6
<i>Parallel Lines</i>	77.8	76.9	72.2	48.2	68.6	69.4
<i>Color Blending</i>	33.3	63	69.4	48.2	56.5	38.9
<i>Pie</i>	72.2	80.6	77.8	45.4	63.9	58.3
<i>Thickness-Shade</i>	64.8	65.8	67.6	40.8	72.2	40.7
<i>Side-by-Side</i>	76.9	80.6	70.3	41.7	65.7	72.2

Task 2 is a trend identification task of 3 variables that involves multiple contour lines along a directed line. *Pie* and *Side-by-Side* got more accuracy than *Color Blending* and *Thickness-Shade* as they (*Pie* and *Side-by-Side*) had B, C and D encoded in 3 separate positions unlike *Color Blending* and *Thickness-Shade* where B and C are encoded in the same space. The same thing resulted *Parallel Lines* to be faster than *Thickness-Shade*.

Task 3 is another 3 variable trend identification task having only 1 contour involved where the user had to look at a specific contour part clockwise. This 3 variable task was completed significantly faster in *Color Blending* than *Side-by-Side*. One reason might be that an user needs to look at 3 different views in *Side-by-Side* where all the other 4 designs have 1 view only.

Task 4 might be the toughest task of all, where user had to find a specific trend involving 3 variables from 4 given choices and then count them. Our assumption is, because of the difficulty of the task, no significance was found.

Task 5 was easier than others, where user had to compare the four marked contour parts to find the contour part having the maximum/minimum difference of 1 variable, going from its' one marked end to another, in clockwise direction. Though RM ANOVA revealed significance in accuracy, Bonferroni-corrected post-hoc t-tests revealed none. The table 9.5 shows that *Thickness-Shade* had the higher accuracy than others. This might have happened because *Thickness-Shade* have all 3 encoding in different forms- B as width, C as shade and D as another line adjacent to B and C. So, it is easy to focus on 1 variable without getting distracted by another with the same encoding. In completion time, *Pie* seems to take the highest time (table 9.4), which might be due to the encoding type of pie where a certain value increase and decrease requires more attention as all 3 variables are encoded similarly and are close to each other.

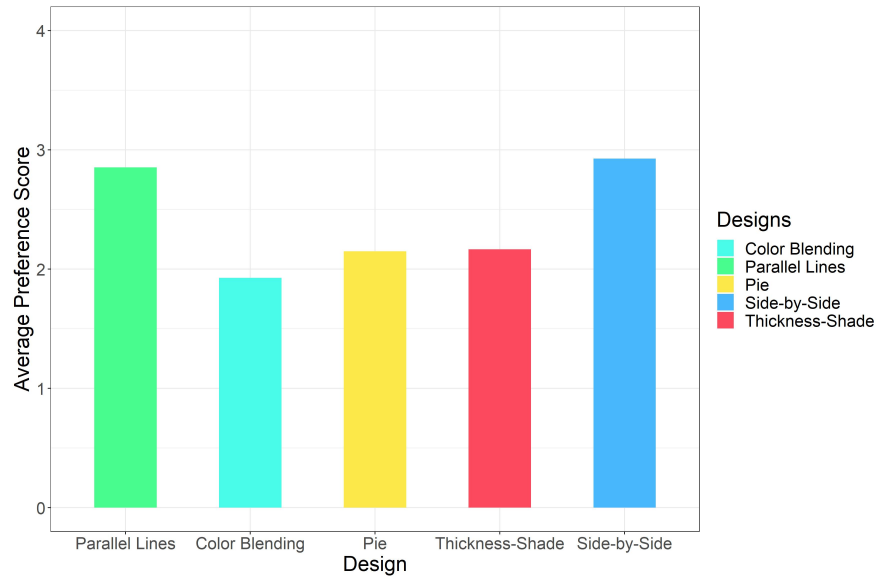
The last task, Task 6 was to identify the contour part having the maximum/minimum difference between 2 variables. In measuring difference, the designs who have the 2 variables (the variables that are mentioned in the task) having similar encoding (i.e. both are encoded using line width or color) are assumed to perform well. Likewise, *Parallel Lines* and *Side-by-Side* significantly had more accuracy than *Color Blending* and *Thickness-Shade*. Moreover, *Parallel Lines* was significantly faster than *Thickness-Shade*.

9.5 Design Recommendations

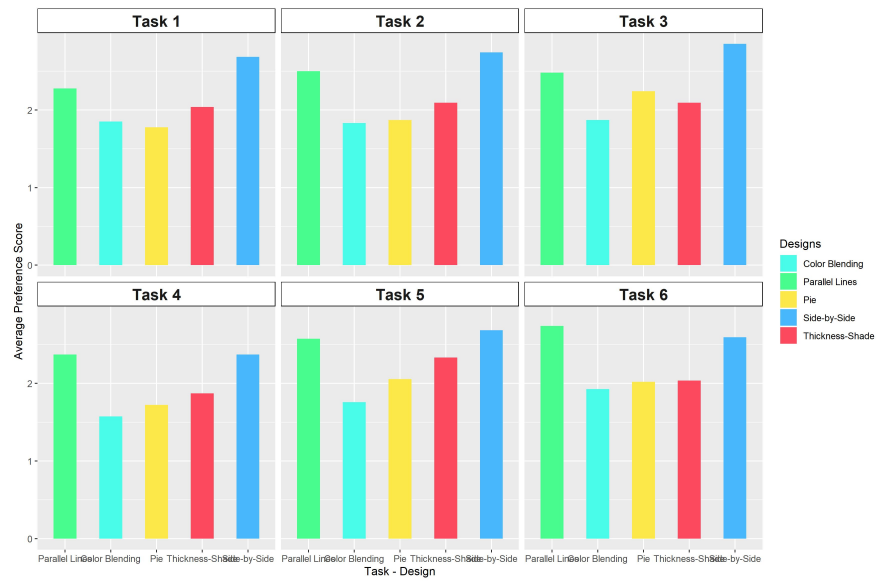
Based on the study results, we formulated a table of design recommendations (Table 9.6) that summarizes the preferred design choices for various tasks over three environments – general use, time-sensitive interpretation, and high accuracy requirements. The table shows some strengths for particular designs that are different from the overall discussion above: for example, if the task requires quick estimation for trends along a contour line, then *Color Blending* and *Thickness-Shade* may be the best design options. Any comparison with ‘[*]’ means that it is statistically significant.

Table 9.6: Design recommendation table

Domain	Environment		
	Time Sensitive	Accuracy Sensitive	General
Compare different contour parts	<i>Parallel Lines</i> (mean-23.8 sec; 47%, 64%[*], and 31% faster than <i>Color Blending</i> , <i>Thickness-Shade</i> [*], and <i>Side-by-Side</i> resp.), <i>Pie</i> (mean-27.6 sec; 38%, 53%[*], and 22% faster than <i>Color Blending</i> , <i>Thickness-Shade</i> [*], and <i>Side-by-Side</i> resp.)	All except <i>Color Blending</i> (mean-33.3%; 57%[*], 54%[*], 49%[*] and 57%[*] less accurate than <i>Parallel Lines</i> [*], <i>Pie</i> [*], <i>Thickness-Shade</i> [*], and <i>Side-by-Side</i> [*] resp.)	All except <i>Color Blending</i>
Search for a trend across contour lines	<i>Parallel Lines</i> (mean-24.9 sec; 28%, and 40%[*] faster than <i>Color Blending</i> and <i>Thickness-Shade</i> [*] resp.), <i>Pie</i> (mean-27.8 sec; 15%, and 25% faster than <i>Color Blending</i> and <i>Thickness-Shade</i> resp.), <i>Side-by-Side</i> (mean-24.9 sec; 18%, and 28% faster than <i>Color Blending</i> , and <i>Thickness-Shade</i> resp.)	<i>Parallel Lines</i> (mean-76.9%; 22% and 17% more accurate than <i>Color Blending</i> and <i>Thickness-Shade</i> resp.), <i>Pie</i> (mean-80.6%; 28%[*] and 23%[*] more accurate than <i>Color Blending</i> [*] and <i>Thickness-Shade</i> [*] resp.), <i>Side-by-Side</i> (mean-80.6%; 28%[*] and 23%[*] more accurate than <i>Color Blending</i> [*] and <i>Thickness-Shade</i> [*] resp.)	<i>Parallel Lines</i> , <i>Side-by-Side</i> , <i>Pie</i>
Search for a trend along a contour line	<i>Color Blending</i> (mean-23.4 sec; 26%, 36%, and 81%[*] faster than <i>Parallel Lines</i> , <i>Pie</i> and <i>Side-by-Side</i> [*] resp.), <i>Thickness-Shade</i> (mean-27.6 sec; 7%, 13%, and 35% faster than <i>Parallel Lines</i> , <i>Pie</i> and <i>Side-by-Side</i> resp.)	All	All
Identify rate of change of a variable along a contour line	All except <i>Pie</i> (mean-35.8 sec; 62%, 39%, 71% and 84% slower than <i>Parallel Lines</i> , <i>Color Blending</i> , <i>Thickness-Shade</i> , and <i>Side-by-Side</i> resp.)	All	All
Identify the value difference on a contour part	<i>Parallel Lines</i> (mean-21.8 sec; 23%, 19% and 46%[*] faster than <i>Color Blending</i> , <i>Pie</i> and <i>Thickness-Shade</i> [*] resp.), <i>Side-by-Side</i> (mean-23.6 sec; 12%, 10% and 35% faster than <i>Color Blending</i> , <i>Pie</i> and <i>Thickness-Shade</i> resp.)	<i>Parallel Lines</i> (mean-69.4%; 78%[*], 19% and 71%[*] more accurate than <i>Color Blending</i> [*], <i>Pie</i> and <i>Thickness-Shade</i> [*] resp.), <i>Side-by-Side</i> (mean-72.2% ; 86%[*], 24% and 77%[*] more accurate than <i>Color Blending</i> [*], <i>Pie</i> and <i>Thickness-Shade</i> [*] resp.),	<i>Parallel Lines</i> , <i>Side-by-Side</i> , <i>Pie</i>



(a) Overall Preference



(b) Taskwise Preference

Figure 9.4: Study 3: design preference survey

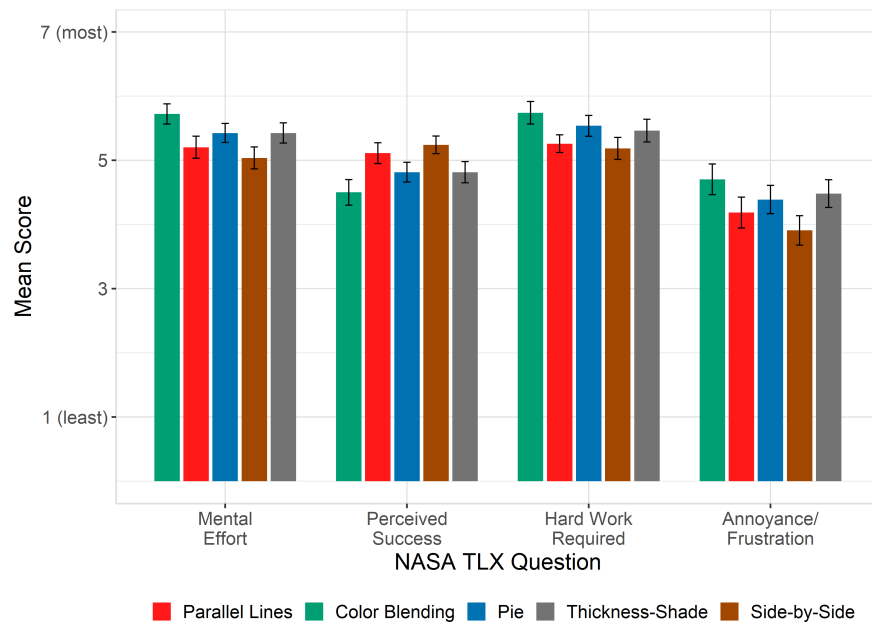


Figure 9.5: Study 3: NASA-TLX scores

CHAPTER 10

STUDY INSIGHTS

10.1 Display Size and Study Type

The three crowdsourced studies in this thesis were run on AMT, which is a well-known platform for running user studies and getting insightful data, which is similar in quality to lab-based study [5, 12, 51]. But, at the same time, the participants completed their tasks on computers having a maximum resolution of 1920×1080 and a minimum of 943×530 , while bigger display sizes have been found performing better in some task completion scenarios [4, 56, 104, 105]. Moreover, there can be other environmental factors influencing user performance, such as the absence of investigators, environmental distractions. So, there remains a chance of getting distracted in online studies [16].

In order to check the effect of study type (lab and online) and display size on accuracy and completion time, we ran Study 1 in a lab setting with a large display.

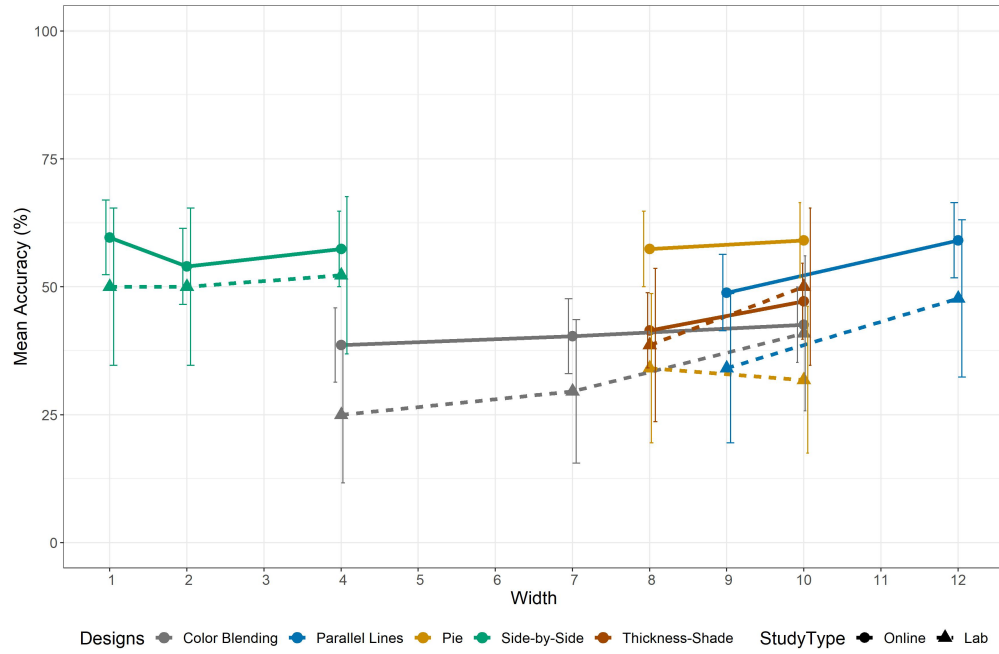
10.2 Study 1 Lab Version

The Study 1 was run both online and at the lab. The display sizes of the users' computers also varied. The lab study was run on a Viewsonic ViewBoard 64" 4K Interactive Panel with a resolution of 3840×2160 .

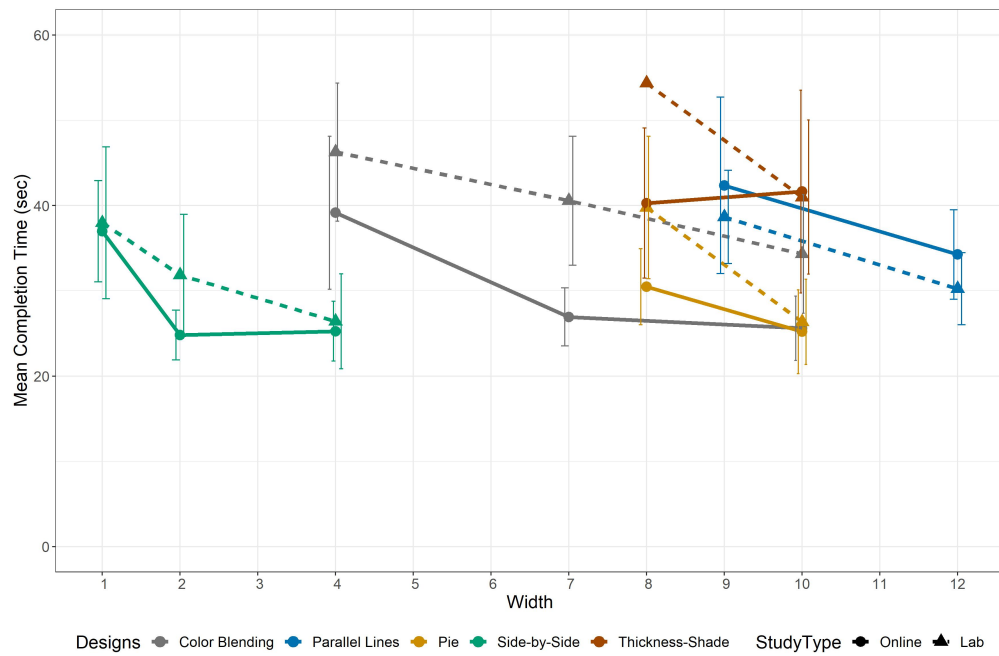
The lab study was run on 12 participants (6 male, 6 female), aged 21-29 (mean age 24.8). The study used a repeated-measures within-participants design. The independent variable i.e. factor was width and the dependent variables were completion time and accuracy. The designs maintained a Latin square order across the participants. After thresholding,

Table 10.1: Lab VS online study hypotheses

ID	Hypothesis
h_6	Bigger screen size will result in better performance than the small screen as the visualization is easier to perceive on bigger screen
h_7	Online study will lead to lower accuracy and higher completion time due to the increased probability of getting distracted during the study and lack of control of the investigator



(a) Accuracy comparison: dashed (solid) lines denote lab (online) study



(b) Completion time comparison

Figure 10.1: Design-wise comparison between lab and online study

the accuracy at 30%, 9 out of 12 participants' data were considered for analysis, while in the online study, we got 44 participants (20 male, 24 female) with a median age range of 31-39.

We had 2 hypotheses for this comparison, as mentioned in Table 10.1.

10.3 Effect of Display Size on Visualization

The mean completion time for the lab study was 37.3 seconds where it took 32.8 seconds for the online study. The accuracy means were 40.3% and 50.4% for the lab study and online study respectively.

The design-wise comparison in Figure 10.1 shows that the accuracy for both display sizes is close to each other, the online study being slightly higher in all widths except for width 10 of *Thickness-Shade*. For completion time, the lab study has higher or equal accuracy except the two widths of *Parallel Lines*. The results show that no major difference was found between large and small display sizes, which does not support the hypothesis h_6 . This implies that the tasks can be completed reasonably well on any display size.

10.4 Lab Study VS Online Study

The 12 participants of the lab study were students from the computer science department who had prior knowledge about visualization. For the online study, the expertise of the participants was completely unknown beforehand and it was possible to run the study on multiple participants concurrently, leading us to gather more participants than the lab study.

After a 30% accuracy filtering, out of 126 responses of the online study, 44 were taken into consideration where it was 9 out of 12 for the lab study. The accuracy filtering filtered out 65% of the data from online responses while it is 25% for the lab study. However, the comparison of filtered responses showed no significant difference between lab and online study. But the fact that online study produced more removable data, partially supports hypothesis h_7 .

The reason might be- the online study was not interactive, and the participants did not feel that much connected and obliged to complete the tasks seriously which is thought to be felt more in the lab study. Moreover, the lab studies were maintained in a calm environment with minimal distraction, like- the participants were requested to keep their cellphones switched off and there was no other person in that room except the investigator. On the other hand, having no such control over the participants' environment might have lead to more inaccurate responses.

10.5 Limitations and Future Work

Our experience with the designs indicates some limitations in the research that provide opportunities for additional study.

First, since we encode the variables at the pixel level, our current implementation does not scale well with large maps. For example, visualizing the temperature trend of 50×50 dataset (a matrix data having 50 points in the row and 50 in the column, each row and column combination making a specific coordinate, i.e. latitude and longitude of the actual geospatial data) onto a 800×800 SVG of Figure 5.4 takes roughly 2.2-2.6 seconds (table 10.2) for an Intel

core i5-8250U 1.60 GHz quad-core CPU with 8 GB RAM where a 200×200 dataset takes 30.2-33 seconds. This time would increase accordingly with the dataset size and SVG dimension, but the most important factor is the number of points that are getting processed for visualization (points found after slicing the contour line segments into smaller pieces and visualizing B, C, and D accordingly). For example, a 160×160 dataset has more points than 200×200 data, which might be due to the underlying data distribution and a number of contour intervals. However, as the points are independent of one another, rendering techniques using GPU acceleration can be used to overcome such an obstacle.

Table 10.2: Completion Time Chart

SVG Dimension	Dataset	No. of points processed	Time (sec)
400×400	50×50	6138	1.3 - 1.8
400×400	120×120	14242	13.5 - 15.2
400×400	160×160	36352	31.7 - 33
400×400	200×200	21792	21.2 - 22.5
800×800	50×50	11609	2.2 - 2.6
800×800	120×120	23811	19.5 - 22.9
800×800	160×160	59471	47.4 - 53.2
800×800	200×200	32738	30.2 - 33

Second, as the number of contour intervals grows, the contours lines may sometimes overlap. Therefore, finding an adaptive choice of the number of contour thresholds can be a valuable avenue of future research.

Third, using existing contours as the basis for additional variables is limited by the density of these contours. Further work is needed on how to represent variables in areas with few contours: for example, adaptive choice of contour intervals could be used to achieve an optimum contour density across the map, or glyph-based techniques could be used to show important changes that occur between contours where the density is less.

Fourth, since we used colors in many of our designs, the interpretability of the designs could depend on the background map color and texture. Therefore, real-world deployments of our technique will benefit from methods that tune color choices to the background map, or by implementing controls so that users can choose the opacity of the background map.

Fifth, the notion of reducing visual clutter and map overlap with our designs were not quantified through any dedicated study. Therefore, a study comparing the designs with the contemporary visualization approaches can be done in the future, to reveal quantitative data.

In addition, we shifted to an online study due to the COVID-19 situation after the lab study version of Study 1. While the crowdsourced surveys have been found to be useful in gaining insights about our designs, additional controlled studies in the lab as well as focus groups with meteorological experts could provide more information. For example, the use of eye-tracking for our approach would give more detail on how users interact visually with the different designs.

Given the complex interaction among various factors that we observed, in-depth observation of the use of the designs in realistic tasks could help better understand some of the effects and interactions.

We plan to apply our designs to different real-world datasets and explore different contour-based tasks and scenarios that can be used in real geospatial settings.

Finally, an interactive system can be built, which will provide users with control over the encoding parameters (such as minimum/maximum thickness, color hues and shades, number of contour thresholds, pie radius, color blending schemes) and will also suggest a set of optimum visualization choices based the environment criteria set by the user.

While we restricted participation from North America to control variability that might appear from diverse cultural backgrounds, a future study with diverse participant demographics can help analyzing the generalizability of our results. Way of data interpretation, design preference, task completion strategy and user performance might vary for participants with different region of residence, occupation, level of expertise and knowledge.

CHAPTER 11

CONCLUSION

Contour plots are widely used, but standard techniques for adding multivariate visualizations onto these plots can clutter the display. To address this problem, we explored how contour lines on a geospatial map can be stylized to encode other attributes in the data. Such a multivariate representation can reduce visual clutter by leveraging the existing contour line space. We designed five types of visual encoding and examined how various contour parameters such as width and contour levels influence task performances. Our crowdsourced study results showed that participants were able to perform a variety of multivariate data analysis tasks with reasonable accuracy, which reveals the potential of our approach.

Study 1 was run to check the effect of contour width on the designs' interpretability and task completion performance. The study was run on 63 participants, each of them completed 57 tasks on various width alternatives. The result revealed that higher width leads to improved completion time in some designs. And also, slightly reduced width is viable for scenarios where the visibility of the background is critical for the accuracy of the task.

Study 2 was aimed to get the idea of how the number of contour lines can affect the performance. Each of the 68 participants completed 60 tasks along with background icon counting tasks. The results revealed that a lower number of contour lines reduces visual clutter, thus improves task performance.

Finally, we ran Study 3 to get some environment and task domain-specific design recommendations and also check the generalizability of the results from synthetic to real-world dataset. In Study 3, we used both synthetic and real-life datasets unlike Study 1 and 2, where we used only synthetic data. Each of the 78 participants completed 60 tasks. In Study 1 and 2, there were 4 unique task types where Study 3 had 6 task types. The results supported the generalizability of the datasets. Each design came out as recommended in at least one scenario in the design recommendation table. Out of the 5 designs, *Parallel Lines* and *Side-by-Side* performed the best and were preferred the most by the users in the design preference survey.

The lab vs online study comparison results partially supported one of the two hypotheses, but an increased number of participants for the lab study with a bigger display has the potential to help gain more insights. And there are several future works to address the limitations of the work. One important future work would be adding meteorological experts to the design study to reveal more task-domain-specific pros and cons of each design and make an interactive system that will suggest a user a set of optimum designs to work with, under a given environment. The main objective of the thesis was to explore the idea of leveraging contour lines for visualization. Reducing visual clutter and background map overlap were some natural aesthetics associated with it. In Study 1 and 2, we explored the effect of line width and a

number of intervals on background map related tasks (icon counting task) that gives us some insights about map overlap and clutter effects. We only examined the visual clutter in the light of an icon counting task. However, quantitative measures such as total visible area and various contour density measures could reveal more insights into the clutter problem. Another dedicated study on them has the potential to reveal more information in the future.

REFERENCES

- [1] Natalia Adrienko and Gennady Adrienko. Spatial generalization and aggregation of massive movement data. *IEEE Transactions on visualization and computer graphics*, 17(2):205–219, 2010.
- [2] Daniel Archambault, Helen Purchase, and Bruno Pinaud. Animation, small multiples, and the effect of mental map preservation in dynamic graphs. *IEEE transactions on visualization and computer graphics*, 17(4):539–552, 2010.
- [3] Benjamin Bach, Charles Perin, Qiuyuan Ren, and Pierre Dragicevic. Ways of Visualizing Data on Curves. In *TransImage 2018 - 5th Biennial Transdisciplinary Imaging Conference*, pages 1–14, Edinburgh, United Kingdom, April 2018.
- [4] Robert Ball and Chris North. Effects of tiled high-resolution display on basic visualization and navigation tasks. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '05, page 1196–1199, New York, NY, USA, 2005. Association for Computing Machinery.
- [5] Christoph Bartneck, Andreas Duenser, Elena Moltchanova, and Karolina Zawieska. Comparing the similarity of responses received from studies in amazon’s mechanical turk to studies conducted online and with direct recruitment. *PloS one*, 10(4):e0121595, 2015.
- [6] Rita Borgo, Johannes Kehrler, David HS Chung, Eamonn Maguire, Robert S Laramée, Helwig Hauser, Matthew Ward, and Min Chen. Glyph-based visualization: Foundations, design guidelines, techniques and applications. In *Eurographics (STARs)*, pages 39–63, 2013.
- [7] David Borland and Russell M Taylor II. Rainbow color map (still) considered harmful. *IEEE computer graphics and applications*, 27(2):14–17, 2007.
- [8] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. D³ data-driven documents. *IEEE transactions on visualization and computer graphics*, 17(12):2301–2309, 2011.
- [9] Lawrence A Bruckner. On chernoff faces. In *Graphical representation of multivariate data*, pages 93–121. Elsevier, 1978.
- [10] Taylour G Burton, Hanadi S Rifai, Zacariah L Hildenbrand, Doug D Carlton Jr, Brian E Fontenot, and Kevin A Schug. Elucidating hydraulic fracturing impacts on groundwater quality using a regional geospatial statistical modeling approach. *Science of the Total Environment*, 545:114–126, 2016.
- [11] Jianhua Cao, Yang Yue, Kunyu Zhang, Jucheng Yang, and Xiankun Zhang. Subsurface channel detection using color blending of seismic attribute volumes. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 8(12):157–170, 2015.
- [12] Krista Casler, Lydia Bickel, and Elizabeth Hackett. Separate but equal? a comparison of participants and data gathered via amazon’s mturk, social media, and face-to-face behavioral testing. *Computers in human behavior*, 29(6):2156–2160, 2013.
- [13] Andrej Cedilnik and Penny Rheingans. Procedural annotation of uncertain information. In *Proceedings Visualization 2000. VIS 2000 (Cat. No. 00CH37145)*, pages 77–84. IEEE, 2000.
- [14] Remco Chang, Caroline Ziemkiewicz, Roman Pyzh, Joseph Kielman, and William Ribarsky. Learning-based evaluation of visual analytic systems. In *Proceedings of the 3rd BELIV’10 Workshop: BEyond time and errors: novel evaluation methods for Information Visualization*, pages 29–34, 2010.

- [15] Herman Chernoff. The use of faces to represent points in k-dimensional space graphically. *Journal of the American statistical Association*, 68(342):361–368, 1973.
- [16] Janelle H Cheung, Deanna K Burns, Robert R Sinclair, and Michael Sliter. Amazon mechanical turk in organizational psychology: An evaluation and practical recommendations. *Journal of Business and Psychology*, 32(4):347–361, 2017.
- [17] Luca Chittaro. Visualizing information on mobile devices. *Computer*, 39(3):40–45, 2006.
- [18] Sidonie Christophe, Bertrand Duménieu, Jérémie Turbet, Charlotte Hoarau, Nicolas Mellado, Jérémie Ory, Hugo Loi, Antoine Masse, Benoit Arbelot, Romain Vergne, et al. Map style formalization: Rendering techniques extension for cartography. In *Expressive 2016 the joint symposium on computational aesthetics and sketch-based interfaces and modeling and non-photorealistic animation and rendering*, pages 59–68. The Eurographics Association, 2016.
- [19] David HS Chung. *High-dimensional glyph-based visualization and interactive techniques*. Swansea University (United Kingdom), 2014.
- [20] David HS Chung, Daniel Archambault, Rita Borgo, Darren J Edwards, Robert S Laramee, and Min Chen. How ordered is it? on the perceptual orderability of visual channels. In *Computer Graphics Forum*, volume 35, pages 131–140. Wiley Online Library, 2016.
- [21] David HS Chung, Philip A Legg, Matthew L Parry, Rhodri Bown, Iwan W Griffiths, Robert S Laramee, and Min Chen. Glyph sorting: Interactive visualization for multi-dimensional data. *Information Visualization*, 14(1):76–90, 2015.
- [22] JH Clark. The ishihara test for color blindness. *American Journal of Physiological Optics*, 1924.
- [23] William S Cleveland and Robert McGill. Graphical perception: Theory, experimentation, and application to the development of graphical methods. *Journal of the American statistical association*, 79(387):531–554, 1984.
- [24] Andy Cockburn, Amy Karlson, and Benjamin B Bederson. A review of overview+ detail, zooming, and focus+ context interfaces. *ACM Computing Surveys (CSUR)*, 41(1):1–31, 2009.
- [25] Edward Colet and Doris Aaronson. Visualization of multivariate data: Human-factors considerations. *Behavior Research Methods, Instruments, & Computers*, 27(2):257–263, 1995.
- [26] National Research Council. *Sea-Level Rise for the Coasts of California, Oregon, and Washington: Past, Present, and Future*. The National Academies Press, Washington, DC, 2012.
- [27] Jonathan Cox, Donald House, and Michael Lindell. Visualizing uncertainty in predicted hurricane tracks. *International Journal for Uncertainty Quantification*, 3(2), 2013.
- [28] Roger Crawfis and Michael J Allison. A scientific visualization synthesizer. In *IEEE Visualization*, pages 262–267, 1991.
- [29] Joseph L. Crockett and A. Leroy Westerling. Greater temperature and precipitation extremes intensify western u.s. droughts, wildfire severity, and sierra nevada tree mortality. *Journal of Climate*, 31(1):341–354, 2018.
- [30] Kevin Crowston. Amazon mechanical turk: A research tool for organizations and information systems scholars. In Anol Bhattacharjee and Brian Fitzgerald, editors, *Shaping the Future of ICT Research. Methods and Approaches*, pages 210–221. Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.
- [31] Michael John De Smith, Michael F Goodchild, and Paul Longley. *Geospatial analysis: a comprehensive guide to principles, techniques and software tools*. Troubador publishing ltd, 2007.
- [32] Danny Dorling, Anna Barford, and Mark Newman. Worldmapper: the world as you’ve never seen it before. *IEEE transactions on visualization and computer graphics*, 12(5):757–764, 2006.
- [33] Selan Dos Santos and Ken Brodrie. Gaining understanding of multivariate and multidimensional data through visualization. *Computers & Graphics*, 28(3):311–325, 2004.

- [34] Richard Dunn. A dynamic approach to two-variable color mapping. *The American Statistician*, 43(4):245–252, 1989.
- [35] David A Ellsworth, Christopher E Henze, and Bron C Nelson. Interactive visualization of high-dimensional petascale ocean data. In *2017 IEEE 7th Symposium on Large Data Analysis and Visualization (LDAV)*, pages 36–44. IEEE, 2017.
- [36] Thomas A Foley and David A Lane. Visualization of irregular multivariate data. In *Proceedings of the First IEEE Conference on Visualization: Visualization90*, pages 247–254. IEEE, 1990.
- [37] Georg Fuchs and Heidrun Schumann. Visualizing abstract data on maps. In *Proceedings. Eighth International Conference on Information Visualisation, 2004. IV 2004.*, pages 139–144. IEEE, 2004.
- [38] Raphael Fuchs and Helwig Hauser. Visualization of multi-variate scientific data. In *Computer Graphics Forum*, volume 28, pages 1670–1690. Wiley Online Library, 2009.
- [39] E Bruce Goldstein. *Cognitive psychology: Connecting mind, research and everyday experience*. Nelson Education, 2014.
- [40] Jochen Görtler, Christoph Schulz, Daniel Weiskopf, and Oliver Deussen. Bubble treemaps for uncertainty visualization. *IEEE transactions on visualization and computer graphics*, 24(1):719–728, 2017.
- [41] Randal Greene, Rodolphe Devillers, Joan Luther, and Brian Eddy. Gis-based multiple-criteria decision analysis. *Geography Compass*, 5:412 – 432, 06 2011.
- [42] Diansheng Guo, Mark Gahegan, Alan M. MacEachren, and Biliang Zhou. Multivariate analysis and geovisualization with an integrated geographic knowledge discovery approach. *Cartography and Geographic Information Science*, 32(2):113–132, 2005. PMID: 19960118.
- [43] Haleh Hagh-Shenas, Sunghee Kim, Victoria Interrante, and Christopher Healey. Weaving versus blending: a quantitative assessment of the information carrying capacities of two alternative methods for conveying multivariate data with color. *IEEE transactions on visualization and computer graphics*, 13(6):1270–1277, 2007.
- [44] Sandra G Hart and Lowell E Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. In *Advances in psychology*, volume 52, pages 139–183. Elsevier, 1988.
- [45] Christopher Healey and James Enns. Attention and visual memory in visualization and computer graphics. *IEEE transactions on visualization and computer graphics*, 18(7):1170–1188, 2011.
- [46] Christopher Healey, Sarat Kocherlakota, Vivek Rao, Reshma Mehta, and Robert St Amant. Visual perception and mixed-initiative interaction for assisted visualization design. *IEEE transactions on visualization and computer graphics*, 14(2):396–411, 2008.
- [47] Christopher G Healey and James T Enns. Large datasets at a glance: Combining textures and colors in scientific visualization. *IEEE transactions on visualization and computer graphics*, 5(2):145–167, 1999.
- [48] Christopher G Healey and James T Enns. Perception and painting: A search for effective, engaging visualizations. *IEEE Computer Graphics and Applications*, 22(2):10–15, 2002.
- [49] Christopher G Healey et al. Perception in visualization. *Retrieved February*, 10:2008, 2007.
- [50] Christopher G Healey, Laura Tateosian, James T Enns, and Mark Remple. Perceptually based brush strokes for nonphotorealistic visualization. *ACM Transactions on Graphics (TOG)*, 23(1):64–96, 2004.
- [51] Jeffrey Heer and Michael Bostock. Crowdsourcing graphical perception: using mechanical turk to assess visualization design. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 203–212, 2010.
- [52] Mario Hlawitschka, Gerik Scheuermann, and Bernd Hamann. Interactive glyph placement for tensor fields. In *International Symposium on Visual Computing*, pages 331–340. Springer, 2007.

- [53] Charlotte Hoarau, Sidonie Christophe, and Sébastien Mustière. Mixing, blending, merging or scrambling topographic maps and orthoimagery in geovisualizations. In *International Cartographic Conference*, 2013.
- [54] Donald H House, Alethea S Bair, and Colin Ware. An approach to the perceptual optimization of complex visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 12(4):509–521, 2006.
- [55] Xiaoke Huang, Ye Zhao, Chao Ma, Jing Yang, Xinyue Ye, and Chong Zhang. Trajgraph: A graph-based visual analytics approach to studying urban network centralities using taxi trajectory data. *IEEE transactions on visualization and computer graphics*, 22(1):160–169, 2015.
- [56] Mikkel R Jakobsen and Kasper Hornbæk. Interactive visualizations on large and small displays: The interrelation of display size, information space, and scale. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2336–2345, 2013.
- [57] Daniel A Keim. Visual techniques for exploring databases. In *Knowledge Discovery in Databases (KDD'97)*, 1997.
- [58] Daniel A Keim and H-P Kriegel. Visualization techniques for mining large databases: A comparison. *IEEE Transactions on knowledge and data engineering*, 8(6):923–938, 1996.
- [59] SungYe Kim, Ross Maciejewski, Abish Malik, Yun Jang, David S Ebert, and Tobias Isenberg. Bristle maps: A multivariate abstraction technique for geovisualization. *IEEE Transactions on Visualization and Computer Graphics*, 19(9):1438–1454, 2013.
- [60] Gordon Kindlmann and Carl-Fredrik Westin. Diffusion tensor visualization with glyph packing. *IEEE transactions on visualization and computer graphics*, 12(5):1329–1336, 2006.
- [61] Robert M Kirby, Haralambos Marmanis, and David H Laidlaw. Visualizing multivalued data from 2d incompressible flows using concepts from painting. In *Proceedings Visualization'99 (Cat. No. 99CB37067)*, pages 333–540. IEEE, 1999.
- [62] Robert Kosara and Caroline Ziemkiewicz. Do mechanical turks dream of square pie charts? In *Proceedings of the 3rd BELIV'10 Workshop: Beyond time and errors: Novel evaluation methods for information visualization*, pages 63–70, 2010.
- [63] Philip A Legg, David HS Chung, Matthew L Parry, Mark W Jones, Rhys Long, Iwan W Griffiths, and Min Chen. Matchpad: interactive glyph-based visualization for real-time sports performance analysis. In *Computer Graphics Forum*, volume 31, pages 1255–1264. Wiley Online Library, 2012.
- [64] Anna Leonowicz. Two-variable choropleth maps as a useful tool for visualization of geographical relationship. *Geografija*, 42:33–37, 2006.
- [65] Sharon Lin, Julie Fortuna, Chinmay Kulkarni, Maureen Stone, and Jeffrey Heer. Selecting semantically-resonant colors for data visualization. In *Computer Graphics Forum*, volume 32, pages 401–410. Wiley Online Library, 2013.
- [66] Fang Liu and Rosalind W Picard. Periodicity, directionality, and randomness: Wold features for image modeling and retrieval. *IEEE transactions on pattern analysis and machine intelligence*, 18(7):722–733, 1996.
- [67] Brent M Lofgren. Sensitivity of land–ocean circulations, precipitation, and soil moisture to perturbed land surface albedo. *Journal of climate*, 8(10):2521–2542, 1995.
- [68] VLK Lou, TE Mitchell, and Arthur H Heuer. Graphical displays of the thermodynamics of high-temperature gas-solid reactions and their application to oxidation of metals and evaporation of oxides. *Journal of the American Ceramic Society*, 68(2):49–58, 1985.
- [69] Linlin Lu and Huadong Guo. Visualization of a digital elevation model. *Data Science Journal*, 6:481–484, 2007.
- [70] Alan M MacEachren. Visualizing uncertain information. *Cartographic perspectives*, (13):10–19, 1992.
- [71] Alan M MacEachren. *How maps work: representation, visualization, and design*. Guilford Press, 2004.

- [72] Jock Mackinlay. Automating the design of graphical presentations of relational information. *Acm Transactions On Graphics (Tog)*, 5(2):110–141, 1986.
- [73] Carsten Maple. Geometric design and space planning using the marching squares and marching cube algorithms. In *2003 international conference on geometric modeling and graphics, 2003. Proceedings*, pages 90–95. IEEE, 2003.
- [74] Joel P Martin, J Edward Swan, Robert J Moorhead, Zhanping Liu, and Shangshu Cai. Results of a user study on 2d hurricane visualization. In *Computer Graphics Forum*, volume 27, pages 991–998. Wiley Online Library, 2008.
- [75] Liam McNabb and Robert S Laramée. Multivariate maps—a glyph-placement algorithm to support multivariate geospatial visualization. *Information*, 10(10):302, 2019.
- [76] Sumedh Yamaji Mhaske and Deepankar Choudhury. Geospatial contour mapping of shear wave velocity for mumbai city. *Natural Hazards*, 59(1):317–327, 2011.
- [77] Mozilla. mix-blend-mode- mdn web docs. [link available here](#).
- [78] Sabrina Nusrat, Muhammad Jawaherul Alam, Carlos Scheidegger, and Stephen Kobourov. Cartogram visualization for bivariate geo-statistical data. *IEEE transactions on visualization and computer graphics*, 24(10):2675–2688, 2018.
- [79] David O’Sullivan, Alastair M. Morrison, and John Shearer. Using desktop GIS for the investigation of accessibility by public transport: an isochrone approach. *International Journal of Geographical Information Science*, 14(1):85–104, 2000.
- [80] Alex Pang et al. Visualizing uncertainty in geo-spatial data. In *Proceedings of the workshop on the intersections between geospatial information and information technology*, volume 10, page 3823. National Research Council Arlington, VA, 2001.
- [81] Amanda Parker, Timothy J Bussey, and Edward L Wilding. *The cognitive neuroscience of memory: encoding and retrieval*, volume 1. Psychology Press, 2005.
- [82] Michael S Patterson. *Multivariate Spatio-temporal Visualization of Socio-economic Indicators Using Geographic Maps*. University of California, Santa Cruz, 2008.
- [83] Charles Perin, Tiffany Wun, Richard Pusch, and Sheelagh Carpendale. Assessing the graphical perception of time and speed on 2d+ time trajectories. *IEEE transactions on visualization and computer graphics*, 24(1):698–708, 2017.
- [84] Timo Ropinski, Steffen Oeltze, and Bernhard Preim. Survey of glyph-based visualization techniques for spatial multivariate medical data. *Computers & Graphics*, 35(2):392–401, 2011.
- [85] Timo Ropinski and Bernhard Preim. Taxonomy and usage guidelines for glyph-based medical visualization. In *SimVis*, volume 522, pages 121–138, 2008.
- [86] Poonam Shanbhag, Penny Rheingans, et al. Temporal visualization of planning polygons for efficient partitioning of geo-spatial data. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.*, pages 211–218. IEEE, 2005.
- [87] Gaurav Sharma and Raja Bala. *Digital color imaging handbook*. CRC press, 2017.
- [88] Haleh H Shenan and Victoria Interrante. Compositing color with texture for multi-variate visualization. In *Proceedings of the 3rd international conference on computer graphics and interactive techniques in Australasia and South East Asia*, pages 443–446, 2005.
- [89] W. Shi, P. Fisher, and M.F. Goodchild. *Spatial Data Quality*. CRC Press, 2002.

- [90] Jeff A. Stuart, Joseph Jaquish, Scott Bassett, Frederick C. Harris Jr., and William R. Sherman. An interactive visualization method for integrating digital elevation models and geographic information systems vector layers. In George Bebis, Richard D. Boyle, Darko Koracin, and Bahram Parvin, editors, *Proceedings of the First International Symposium on Advances in Visual Computing (ISVC)*, volume 3804 of *LNCS*, pages 553–561. Springer, 2005.
- [91] Martin Suntinger, Hannes Obwegger, Josef Schiefer, and M Eduard Groller. The event tunnel: Interactive visualization of complex event streams for business process pattern analysis. In *2008 IEEE Pacific Visualization Symposium*, pages 111–118. IEEE, 2008.
- [92] Hideyuki Tamura, Shunji Mori, and Takashi Yamawaki. Textural features corresponding to visual perception. *IEEE Transactions on Systems, man, and cybernetics*, 8(6):460–473, 1978.
- [93] Dejan Todorovic. Gestalt principles. *Scholarpedia*, 3(12):5345, 2008.
- [94] Xin Tong, Cheng Li, and Han-Wei Shen. Glyphlens: View-dependent occlusion management in the interactive glyph visualization. *IEEE transactions on visualization and computer graphics*, 23(1):891–900, 2016.
- [95] Jarke J van Wijk and Alexandru Telea. *Enridged contour maps*. IEEE, 2001.
- [96] AHC vanderHeijden. Perception for selection, selection for action, and action for perception, 1996.
- [97] Tim Vasquez. *Weather forecasting handbook*.
- [98] Zuchao Wang, Min Lu, Xiaoru Yuan, Junping Zhang, and Huub Van De Wetering. Visual traffic jam analysis based on trajectory data. *IEEE transactions on visualization and computer graphics*, 19(12):2159–2168, 2013.
- [99] Matthew O Ward. Multivariate data glyphs: Principles and practice. In *Handbook of data visualization*, pages 179–198. Springer, 2008.
- [100] Colin Ware. *Information Visualization, Second Edition: Perception for Design (Interactive Technologies)*. 04 2004.
- [101] Keqin Wu and Song Zhang. Visualizing 2d scalar fields with hierarchical topology. In Shixia Liu, Gerik Scheuermann, and Shigeo Takahashi, editors, *2015 IEEE Pacific Visualization Symposium, PacificVis 2015, Hangzhou, China, April 14-17, 2015*, pages 141–145. IEEE Computer Society, 2015.
- [102] www.colorhexa.com. Website for color choice. [link available here](http://www.colorhexa.com).
- [103] www.w3schools.com. mix-blend-mode- editor. [link available here](http://www.w3schools.com).
- [104] B. Yost and C. North. The perceptual scalability of visualization. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):837–844, 2006.
- [105] Beth Yost, Yonca Hacıahmetoglu, and Chris North. Beyond visual acuity: The perceptual scalability of information visualizations for large displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI ’07, page 101–110, New York, NY, USA, 2007. Association for Computing Machinery.
- [106] Jiawei Zhang, Abish Malik, Benjamin Ahlbrand, Niklas Elmqvist, Ross Maciejewski, and David S Ebert. Topogroups: Context-preserving visual illustration of multi-scale spatial aggregates. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 2940–2951, 2017.
- [107] Jiawei Zhang, Chittayong Surakitbanharn, Niklas Elmqvist, Ross Maciejewski, Zhenyu Qian, and David S Ebert. Topotext: Context-preserving text data exploration across multiple spatial scales. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2018.